

THE PATHAGORAS SYSTEM

© 2015 Innovative Software Products of Virginia, LLC

Document Assembly
Document Automation
Document Management

The Pathagoras System

**Document Assembly
Document Automation
Document Management**

by Innovative Software Products of Virginia, LLC

Pathagoras is a different kind of document assembly/document automation system. It shuns hidden fields and codes. It stores source documents in standard folders and files, without encryption and without restrictions which might otherwise limit use of these documents to just the program. It operates purely on plain text -- the kind of text that a typical word processing operator with no programming experience can easily compose and edit.

The program augments what you have and what you know. It does not attempt to replace it. It implements Word concepts with which you are already familiar, but does so in a new and faster way.

There are few new or foreign concepts introduced. Most are simply standard Word concepts elevated to provide document automation not seen before in a plain-text based system.

We hope that you like what you see. And if you have any questions or concerns, we hope that you will contact us and let us know.

© 2015 Innovative Software Products of Virginia, LLC

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

As an exception, each licensed user of Pathagoras may create (for his or her personal or office use) a single printed copy of this Manual for each license owned.

All references in this Manual to "Word" refer to Microsoft® Word, a product of the Microsoft® Corporation. All references to Windows refer to the Microsoft® Windows® operating system, a product of the Microsoft® Corporation.

Other products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: November 2015 in Yorktown, Virginia

Table of Contents

Foreword	xvii
Part I Introduction	1
1 Definitions.....	5
2 Pathagoras in 1...2...3 steps.....	9
3 Design & Work Flow Concepts.....	11
4 Processing vs. Personalizing.....	17
5 Moving from basic to advanced.....	20
6 File and Folder 'Pointers'.....	21
7 Advantages of Plain Text Document Assembly.....	23
8 Document Assembly vs. Document Management.....	24
9 Beyond 'Major' Projects.....	25
10 Help Connections.....	26
11 Upgrading Pathagoras.....	26
Part II Getting Started	29
1 Download and Install.....	30
2 Prepare Folder(s) for Document Assembly.....	31
3 Creating a New Library.....	32
4 Shelving Your First Book.....	35
5 'Pathagorizing' Your First Form.....	37
6 Installation Checklist.....	38
Part III Document Assembly	41
1 The Process, Technique 1.....	43
2 The Process, Technique 2.....	45
3 Libraries & Books: Concepts and Definitions.....	46
4 Libraries & Books Screen_2.....	47
5 'FreeStyle' Display.....	51
6 Clause Selection Screen.....	52
Files Included in Display	53
Buttons and Functions	56
Add and Remove.....	56
Show All.....	57
Select All.....	57
'Rearrange' arrows	57
Forms 'Packages'.....	58
Widen Screen.....	59
Assemble vs. Insert.....	60
View/Edit, etc.	60

	Names vs. Subjects	60
	Linear vs. Stacked.....	61
	Preview	62
	Expand (Clause Set).....	62
	Tree Service.....	62
	Double Click Tools	63
	Assembly Options.....	64
	Add 'Hidden' Clause Names to text	65
	Master Clause Table	66
	Document Assembly Issues	67
7	Document Assembly 'Books'.....	68
	Folder of Documents	69
	Glossaries	70
	The 'Technical Side' to Glossaries	71
	Glossary vs. Folder of Terms	75
	Adding ('Shelving') a Book	76
	Quick Set.....	77
	Pointing/Repoining via Settings Screen.....	78
	Removing ('unshelving') a Book	81
8	Creating a New Library.....	82
9	Deleting a Library.....	85
10	Settings Screen.....	86
	Assigning a Template	88
	Assigning an IDB Mask	88
	Assigning a Prefix	88
	Show All Files in Folder	89
11	SuperBooks.....	90
	SuperFolder	90
	SuperGlossary	91
	Adding Terms to a SuperBook	92
	Compare to Prefix/Suffix Naming	93
	Compare to AutoCorrect/AutoText	93
12	Mouseless Assembly.....	94
13	Order of Search.....	95
14	Templates.....	96
	The Templates Screen	97
	Creating a Template	98
	'StartHere' Location	100
	Assign Template to Book	101
	Exempting Document from Assignment	102
	Assign Template to Document	102
	Paragraph Styles	103
15	<<Document Name>> Reference.....	104
16	Tree Service.....	106
17	Tables and Pathagoras.....	110
18	'Tag Along' Clauses.....	112

Part IV 'Pathagorizing' Your Documents (Variables and Optional Text)

115

1	Variables - Definition.....	117
2	Creating Simple Variables.....	117
3	Creating Multiple Choice Variables.....	118
	Groups.....	119
	Titles	121
	Sharing Groups	122
4	'Create Variables' Wizard.....	122
	CVA Creating New Variables	123
	CVA Using Existing Variables	124
	CVA Adding New Variables	126
	CVA Sanitizing Document with +[. . .]	126
5	The 'Intelligent A'.....	127
6	'And/Or', 'A/K/A' and other slashes.....	127
7	Reusing Existing Variables.....	128
8	Drop Down List.....	129
9	Drag and Drop.....	131
10	Pathagorizing using the IDB screen.....	133
11	Covertng HotDocs Documents.....	135
12	Nested Variables.....	135
13	'Emphasis' (bold, ital, etc.).....	137
14	Date, Formula and 'File Name' replacements.....	140
15	Limitations on Variables.....	140
16	<<*Options/Optional*>> Text Blocks.....	141
	'Optional' Text ('Prompts')	144
	'Options' Text ('Prompts')	146
	Radio vs. Multiple Choice	148
	Nesting	148
	Nesting Variables within Options	150
	Groups	151
	"Syntax"	152
	<<*Options*>> using Tables	154
	Create <<*Options*>> Assistant	155
	Options/Optional Text in Headers/Footers	157
	Simple Options.....	158
	Suppress Processing	158
	Negative Optional	160
	Testing & Editing	160
	Administrative Text	162
	Select Optional text via Options	162
17	{Simple Optional} Text Blocks.....	164
	Differences	167
	Converting	168
	Enabling/Disabling	169
	Options (Mixing Simple with Complex)	170
	Negative Simple Optional Text	171
18	Dynamic Creation of Variables.....	171
19	Structure/Integrity Checker.....	173

20	Debugging Variables & Options Text.....	174
21	Changing Variable Names.....	176
22	Anatomy of Variable and Optional Text Characters.....	176
23	Samples DropDown List.....	180
Part V Quick Picks		182
Part VI Instant Database Screen		185
1	The basics.....	186
2	Scan.....	188
3	Completing a Variable.....	191
4	Replacing Variables with Values.....	192
5	Data Records.....	192
6	Augmenting an Existing Record.....	194
7	Variables, Number of.....	195
8	Variables, Length of.....	195
9	Variables, Special.....	197
10	Validating Variables.....	198
11	Math and Date Math.....	199
12	Multiple Choice Selector.....	203
13	Embed Variable Names.....	206
14	AutoScan.....	207
15	Other IDB Functions, Features and Tools.....	208
16	Power Tools.....	210
17	More Tools and Settings.....	212
18	Still More IDB Settings.....	216
19	Other Uses for IDB.....	217
20	Advanced Array.....	218
21	Screen Shots Depicting Screen Features.....	220
Part VII Instant Database Functions		223
1	Calendar and Date Math Features.....	224
2	In-line Math and Date Math.....	227
3	<<Folder>> Function.....	228
4	'=' Equivalency Function (single element).....	230
5	'=' Equivalency Function (*Actors* and their Roles).....	231
6	Concatenation of Variables.....	237
7	<S>pell out Function.....	238
8	<\$>Currency Function.....	239
9	<%> Percent Function.....	241

10	<F>ormat Function.....	242
11	<Fr>action Function.....	242
12	<<Filename>> Function.....	244
13	<P>aragraph Function.....	245
14	<MailTo:> function.....	246
15	<URL:> function.....	247
16	<.x.> Function Reminders.....	248
Part VIII Instant Database Records		249
1	IDB Files Location.....	250
2	Sharing Data with other programs.....	254
3	Archiving IDB Records.....	254
4	Easy As Pie IDB Records.....	254
5	Print IDB Data from Multiple Records.....	254
6	Changing Variable Names in Existing Records.....	256
Part IX Intake Forms		259
1	Instant Database 'Input Form' Masks.....	260
	Creating a Mask	262
	Mask Maker	265
	Making the Mask More Meaningful	267
	Organizing Mask Entries	268
	Assign Mask to Book	269
	Using a Mask to 'Pathagorize' a document	270
	Sharing Masks	271
	Password	271
	Using Excel to Create Input Form	271
	Importing Data Into Mask	272
2	Word-based Intake Forms.....	272
	Simple Two Column Table	273
	'Fancy' Word Intake Forms	274
	Creating 'Fancy' Form.....	274
	Scraping the Data.....	278
	Transmitting the Form.....	281
	Sample 'Fancy' Forms	281
3	Excel Intake Forms.....	283
4	Adobe™ Acrobat™ as Intake Form.....	283
Part X External Database Connections		285
1	Definitions and Concepts.....	286
2	Getting started	287
3	. . . with Outlook.....	290
4	. . . SQLServer and MySql.....	291
5	. . . with Excel.....	294
6	Simple Merge of External Data.....	295

7	Updating Your Data.....	296
8	Create Record via Web-based Intake Form.....	297
9	Using *MultiChoice* Lists.....	297
10	Create Record via External Program.....	297
11	Pairing Table.....	299
12	'Fielded' Database Connections.....	299
Part XI Mail Merge		300
Part XII Clauses: Adding and Editing		305
1	.doc vs. .dot.....	308
2	Glossary vs. Folder of Terms.....	308
3	Via Libraries & Books screen.....	308
4	Via DropDown Lists.....	310
5	Term Works!.....	311
6	Adding Clauses Manually.....	311
	To a folder of clauses	311
	To a Glossary	313
	The Quickest Glossary Ever	314
	Easy As Pi Glossary.....	315
	Glossary from 'Painted' Document	316
7	Clause Conversion Tools.....	318
	Glossary to Folder	318
	Folder to Glossary	318
	WordPerfect to Word Folder	319
8	Names and Subjects.....	320
9	Styles.....	322
10	Automatic Paragraph Numbering.....	324
	General Information	324
	Re-numbering Issues	325
Part XIII A Different Approach		327
Part XIV Document 'Dis'-assembly		329
1	Manually.....	330
2	Using Libraries & Books Screen.....	330
3	Using DropDown List.....	331
4	Bulk Document Dis-assembly.....	332
Part XV Term Works!		333
1	Screen Tour	334
2	Selecting/Refining Scope.....	335
3	Add Clause to Existing Book.....	337
4	Creating New Book.....	339

5	Prefix/Suffix Naming.....	341
6	Bulk Add connections.....	342
7	Show Text.....	343
8	Add text Where?.....	344
9	Setting Term Works! Defaults.....	345
10	Hints and Tips for Document Disassembly.....	345
Part XVI GotForms?		347
1	Standard Screen.....	348
2	Minimized Screen.....	349
3	Show Context.....	350
4	Underlines as Variables.....	351
5	GotForms?: When To Use?.....	352
Part XVII DropDown Lists		355
1	From Document Assembly.....	357
2	From the Clause Selection Screen.....	357
3	'Free Hand'.....	358
4	Adding Content to DropDown Lists.....	359
5	Repointing a DropDown List.....	360
6	Using a List.....	362
7	The DropDown List Panel.....	364
8	Collections.....	369
9	Environments.....	371
10	DDL Variables List.....	372
11	Community DropDown Lists.....	375
12	Debugging the List.....	377
13	WordPerfect, PDF and Image Assembly.....	378
Part XVIII 'Repeat' Blocks		381
1	Simple Repeats.....	382
2	Simple, with incrementing variable.....	383
3	Repeat with !Groups!.....	384
4	Anatomy of the Repeats Block.....	386
5	Series Connectors ('and' or 'or').....	386
6	Repeating Tables & Rows.....	387
7	'Return' the Repeat value.....	389
8	Repeat Settings.....	391
9	'AskRepeat'.....	392
10	Repeat with Options.....	392

11	Repeats Elsewhere.....	395
12	Repeat Alternatives.....	395
13	Repeat Restrictions.....	398
14	Testing Repeat Blocks.....	398
15	Repeat Example.....	399
16	Repeat (a 'one page' lesson).....	401

Part XIX Clause Sets 405

1	Creating Sets from Clause Selection Screen.....	407
2	Create Set from DropDown List.....	409
3	Creating Clause Sets Manually.....	411
4	Editing Clause Sets.....	413
5	Document Assembly with Clause Sets.....	414
6	Expanding Clause Sets.....	415
7	Notes, Hints and Tricks.....	416

Part XX Prefix/Suffix Naming Convention 419

1	What Is It?.....	420
2	Why use it?.....	421
3	Printing a checklist of names and subjects.....	422
4	Prefix Table.....	424
5	Prefix 'Parsing'.....	425
6	What if I change my mind?.....	425
7	Margin List Assembly.....	426

Part XXI Names and Subjects Editor 427

1	Screen Tour	429
2	Simple Application.....	430
3	Activating the Editor.....	430
4	Advanced Application: Prefixes.....	432
5	Advanced Application: Renumbering.....	436
6	Notes and Miscellaneous Items.....	438

Part XXII Right Click Menu 440

Part XXIII Miscellaneous Editing Tools 445

1	Balancing Markers.....	446
2	Color Coding (Paint Markers).....	447
3	Accessing Source Clauses.....	447
4	Search & Replace.....	448
	Search and Replace Form	448

Search and Replace via IDB	451
5 Editing Tips.....	452
6 Wizards and Assistants.....	454
7 Bulk Editing Tools.....	454
Part XXIV Aliases and MultiChoice *Lists*	457
1 Creating Alias *Lists*	460
2 Easy As Pi *Alias* List.....	465
3 Excel Alias *Lists*	466
4 Presenting Alias *Lists*	467
5 Cascading Options.....	468
6 Cascading Options Chart.....	472
7 Cascading Variables.....	473
8 Aliases and !Groups!.....	476
9 Alias Files Location.....	478
10 Aliases from Database.....	479
11 'Virtual' *List*	484
12 Locating, Moving, Pointing (to).....	486
13 Alias Table (Embedded).....	488
14 Sharing Your Lists with the World.....	489
15 Repairing Alias *Lists*	490
Part XXV Document Logic	491
1 Simple Logic.....	493
2 Intermediate Logic.....	495
3 Cascading Logic.....	496
4 Advanced Logic.....	497
5 Interviews.....	498
The <<*Ask* . . *>> Commands	499
<<*Ask*>> Elements	501
<<*AskOptions*>> options	503
<<*AskValue*>> Command	503
<<*If*>> Command	506
<<*Set*>> Command	508
Creation of Interviews	509
Automatic Creation of <<*Ask*>> prompts	509
Logic Assistant.....	510
Limitations	513
Interview Documents	514
Saving <<*Ask* . . . records	516
Recalling <<*Ask* . . . record	516
Print Interview	517
'Case' Logic	518
Requirements	522
Interview Examples	523

Simple Interview Examples.....	523
A Whimsical Example.....	524
Another Whimsical Example.....	525
A 'Legal' Example.....	527
Dissecting an Ask Table.....	529
Dissecting another Ask Table.....	531

Part XXVI Document Management 533

1 Definitions.....	534
2 PathSmart Screen.....	535
SmartPath	536
SuperSmartPaths	537
Navigating the SmartPaths	538
Range and Filter Controls	539
Print Out SmartPaths	539
3 SaveSmart.....	540
Saving a Document	542
Moving a Document	543
SaveAs/MoveAs	544
Deleting a Document	544
The More>> controls	544
Automatic Document Naming	546
Save to PowerPoint	547
Save as Source Clauses	547
4 Creating a New Profile.....	548
5 Deleting a Profile.....	548
6 Adding a SmartPath.....	548
Quick Set	549
Via Settings Screen	549
Removing a SmartPath	549
7 Settings.....	549
8 Mouseless Navigation.....	550
9 QuickLinks.....	550

Part XXVII Main Menu 553

Part XXVIII Network Operations 555

1 The Common Profiles Path.....	557
2 Enabling the Network.....	557
3 Network Features.....	560
Sharing Files	560
Messaging	563
4 Instant Database Issues.....	563
5 Upgrading Pathagoras.....	564
6 Synchronizing Books.....	565
7 'Use Local' Files.....	566

Part XXIX Settings	569
1 Settings Assistant.....	571
2 Utilities & Settings.....	572
Current Settings	572
Upgrades	573
System Tools	574
3 End User Settings.....	575
4 All Settings and Options.....	576
Document Assembly	577
File Locations	578
Separators and Connectors	578
5 Sharing Registry Settings.....	579
6 Registry Settings.....	581
Part XXX HotKeys	589
1 Alt-G.....	590
2 Alt-S.....	591
3 Alt-D.....	591
4 Alt-V.....	591
5 Changing a Hotkey Assignment.....	591
Part XXXI Typing Expander	592
Part XXXII Search Function	593
Part XXXIII Navigating with Pathagoras	597
1 Locating document in a Book.....	598
2 Locating clause in a Glossary.....	599
3 Locate a clause in a DropDown List.....	600
4 Text boxes displaying folders.....	601
Part XXXIV Compatibility Issues	603
1 Worldox®.....	604
2 WordPerfect®.....	605
3 HotDocs®.....	608
Part XXXV Quick Access Toolbar	610
Part XXXVI Excel Connections	613
1 Excel and DropDown Lists.....	614
2 Excel and the Instant Database.....	616
Part XXXVII Comparative Approaches	621

1	Clauses vs. Templates.....	622
2	Naming Your Clauses.....	625
	Prefix/suffix style vs. descriptive names	625
	Manual vs. Auto Naming of final documents	626
3	Naming Your Variables.....	626
	Bracketed vs. Non-bracketed variables	627
4	Multiple choice variables vs. Multichoice *Lists*.....	627
5	<<*Options/Optional*>> text vs. variables.....	628
6	<<*Options*>> vs. Multiple choice variables.....	630
7	<<*Options*>> vs. Clause Selection Screen.....	630
8	<<*Optional*>> vs. Selecting Clauses.....	631
9	Moving Text from Source to Final Product.....	632
	Clause Selection Screen vs. DropDown lists	632
	Clause Selection Screen vs. Clause Sets	632
	Mouseless vs. everything else	633
10	Personalizing Your Documents.....	633
	GotForms? vs. InstantDatabase	633
	InstantDatabase Mask vs. InstantDatabase Scan	633
11	Storing Your Source Documents & Clauses.....	634
	Folders vs. Glossaries	634
	Documents vs. Templates	635
	Mouseless vs. Everything Else 2.....	637
	Using PathSmart Screen vs. Word 2.....	637
	Direct Add vs. Duplicate 2.....	637
	'Highlight & Add' vs. Direct Add 2.....	638
	Bulk Add vs. Instant Book 2.....	638
	SuperBooks vs. other locations 2.....	638
	SuperBooks vs. other locations	639
	Local vs. Network Storage	640
12	Adding Text to Source Books.....	640
	Bulk Add vs. Instant Book	640
	'Highlight & Add' vs. Direct Add	640
	Direct Add vs. Duplicate	641
13	Navigating.....	641
	Using PathSmart Screen vs. Word	641
	Mouseless vs. Everything Else	642
14	Word Equivalents.....	642

Part XXXVIII Adobe™ Acrobat™ Interactivity 643

1	Replacing using [IDB] variables.....	645
2	Adobe and Scanning.....	647
3	Adobe Data Scraping.....	648
4	Adobe PDF 'Assembly'.....	651

Part XXXIX Miscellaneous Matters 653

1	Four-element toolbar.....	654
----------	----------------------------------	------------

2	Repointing after moving files.....	654
3	Two-element Toolbar.....	655
4	Navigation via Textboxes.....	655
5	Other Navigation Tricks.....	656
6	Pointers (in general).....	656
7	Default Settings.....	656
8	Administrative File Locations.....	657
9	System Security.....	660
10	Document Security.....	661
11	Cross-References.....	664
12	Apply 'Wait'.....	665
13	Compare to HotDocs® and others.....	665
14	Compare to Quick Parts.....	669
15	Comments.....	670
16	Backup/Restore.....	671
	Backup	672
	Restore	672
17	'Plain Vanilla' Paragraph Numbering.....	672
	SavingStyles	673
	Using Plain Vanilla	673

Part XL Tips and Tricks 677

1	Baby Steps.....	678
2	Captions and Signatures.....	679
3	Capturing Data Independently of Document.....	680

Part XLI Upgrading Issues 683

1	Upgrading Pathagoras.....	684
2	Upgrading my system or adding new users.....	684
	Upgrading Checklist	686
3	Annual Support Agreement.....	687

Part XLII Support 689

1	Customer Service.....	690
2	Non-technical Support.....	692

Part XLIII Help System Improvement Project 693

Part XLIV Uninstalling Pathagoras 694

Part XLV One Page Quick Start Guide 696

Index**699**

Pythagoras was a 1st Century B.C. Greek philosopher best known for the mathematical formula named after him.

More importantly to this program, Pythagoras believed that the essence of all things is number, and that all things can be expressed numerically. Pathagoras, the program, distills the theory as it might be applied to word processing by assigning complex DOS paths and file names to simple numbers. The original versions of this program (the "PathSmart" module in the current version) is a perfect adaptation of the "Path = Number" concept.

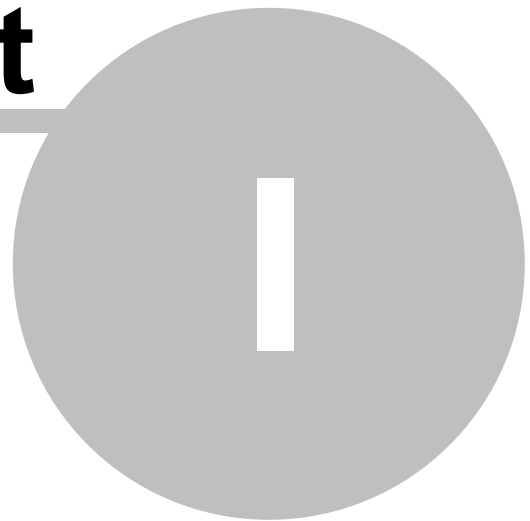
While surely Pythagoras had more global things in mind, this program nevertheless is proud to have this remarkable thinker as its namesake.

By the way, it is pronounced p'-THA-go-rus.

The Pathagoras System

Introduction

Part



1 Introduction



This comprehensive Help System supersedes all of the previously published individual Manuals and User Guides.

This Manual is updated through release 2016 (update September 8, 2015).

See this link for the 2015-2016 updates.

Some functions describe herein operate only with release version 2015 or higher. You can check the version installed on your computer via the Pathagoras 'Main Menu' | 'About' tab.

Like Microsoft, Pathagoras is no longer upgrading Pathagoras for installations on 2002 and 2003 versions of Word.

However, almost all primary functions described in this Manual perform similarly in all versions.

We would greatly appreciate your input as to the usefulness and 'usability' of this Help System.

If you have comments, corrections or suggestions for improvement, please send us an email by clicking [here](#).

If you see an error on a specific page, or wish to offer a comment about a page that you are viewing, click the 'envelope' in the upper right hand corner of that page. That will activate your email program, and address a letter to us with a specific reference to the page you are viewing. Just add your comments and send. Thanks!

Pathagoras™ is a multi-faceted document authoring tool. It is designed to help you create documents of any nature, and to recall them instantly to the editing screen to be personalized for a client or customer. Is well suited whether the project is a complex transaction or a simple letter.

First and foremost, Pathagoras is a 'plain-text based' document assembly and automation system. In just a few clicks, Pathagoras can present a list of forms or clauses on a designated topic. From that list, you can select an appropriate form, group of forms, or a subset of clauses to address your client or customer's need. Pathagoras then quickly assembles those clauses into a complete document.

Secondly, Pathagoras is a database. With a few additional keystrokes, you can assign 'personal' values to replace variables that resided in the source text. You can then easily store and reuse that personal data in future documents containing the same variables.

Pathagoras is also a document *manager*. That means that you can use Pathagoras to help to organize and save documents and to retrieve them for later use.

Here are a few other key features about the program:

- The forms in a 'Pathagorized' system are *your* forms. The language is *your* language.
- There is no pre-packaged text that limits you in your document production. (However, if you already have forms on your computer in which variables are noted between square brackets, those forms are already 'automated documents' as far as Pathagoras is concerned. No further editing is needed.)
- Pathagoras self-generates the data input forms it needs to preserve 'personal' data

(names, addresses, etc.) You do not have to spend any time composing data intake sheets.

- Pathagoras self-generates the Interview forms which can guide you through the selection of "optional" and multiple choice text blocks. There are no complex interview forms that you have to create, and no complex formulas that you have to write to link interview answers to results.
- Pathagoras does not require you to abandon your current files or file structure. Pathagoras easily adapts to the methods that you currently use to store and retrieve documents. Your current folder(s) of forms can remain intact. Each can be used as a Pathagoras 'book' without any special configuration, programming or changes.

These design features set Pathagoras apart from its competitors because of their simplicity of use and ease of implementation.

=====

The majority of the DemoDocuments that accompany the program and the images in this Manual, feature law office forms. However, Pathagoras is a program for *all* offices, *all* disciplines, and for *all* aspect of office operations. Pathagoras is currently used by companies in the following non-law disciplines:

proposal generation (RFPs and Bids)	construction	accounting
financial planning	insurance (claims processing)	software development
medical office	school psychology	home inspection and repair

Pathagoras can be used to enhance any Word operation in any office. Many purchase Pathagoras to assist in drafting contracts, pleadings, reports and proposals. But Pathagoras is also intended to simplify the production of simple letters and day-to-day memos as well. If you can do it in Word, you can do it easier, faster and more accurately using Pathagoras.

See also [Beyond Document Assembly](#) ²⁵

'Plain-text' refers to the structure of the variables and optional text blocks contained within a form.

- A 'plain-text' variable is one that can be composed using standard keyboard-entered text and which you can read and edit along with other text on your screen.
- A plain-text variable is simply a word or phrase enclosed within square brackets.

[This is a variable]

- Optional text is also composed in plain-text. Enclose optional text within "{" and "}" markers. When encountered during an assembly session, the program will ask "Do you want to keep this text?"
- 'Plain-text' is just what it says. It is not a field, a SmartTag or other coded or hidden element that requires a separate program to create or interpret.
- Plain-text does not, however, mean unformatted text. To the contrary, all formatting characteristics that you currently use (bold, italics, indents, etc.) fall within the ambit of 'plain-text.'

Final Introductory comments:

- This Manual is big. Its size reflects the many features of the program.
- Don't try to learn it all. It cannot be done. Take small steps.
- Pathagoras is modular. Everything builds on the next. No step is wasted if you rest without fully completing a project. So if you need to stop, you are already 'ahead' in terms of productivity with whatever you have done.
- On the other hand, if you are every fully 'satisfied' with how you have set up the

program for your office, don't stop. Pull out this Manual and see if there is a 'next step' you might take to make Pathagoras even more productive.

And never hesitate to call us with any questions or suggestions. Our contact information is [right here](#).

Pathagoras, PathSmart, SaveSmart and 'plain-text document assembly' are trade-marks or service marks of Innovative Software Products of VA, LLC, and Roy Lasris.

Pathagoras, PathSmart and SaveSmart programs, interface screens, help system and all other visible and non-visible coding Copyright © 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 and 2011 by Innovative Software Products of Virginia, LLC, and Roy Lasris.

All references in this Manual to "Word" refer to Microsoft® Word, a product of the Microsoft® Corporation. All references to Windows refer to the Microsoft® Windows® operating system, a product of the Microsoft® Corporation.

Other products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this help system, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

1.1 Definitions

Definitions:

[Document Assembly](#):

[library](#)

[book](#)

[clause](#)

[folder of clauses](#)

[package of documents](#)

[glossary](#)

[source](#)

[variable](#)


[Optional text](#)


[Options text](#)

[Repeating text](#)

[Pathagorizing \('neutering'\) a document\)](#)

[Document Disassembly](#)

['Processing' a document](#)  8

['Personalizing' a document](#)  8

[Administrative Text](#)  9

[Document Management:](#)  9

[Profile](#)  9

[SmartPath](#)  9

[SuperSmartPath](#)  9

Document Assembly:

'Document assembly' is the process by which an operator creates an entire document from a variety of component parts and then personalizes that document to meet the needs of the intended recipient.



Included within the scope of the term 'document assembly' are the means by which the source clauses are:

- created
- neutered, and
- assembled.

Pathagoras has adopted a 'library' and 'books' metaphor to describe the manner in which it organizes documents and forms.

Library: A 'library' is the top-most level. It is a collection of up to 10 'books,' along with rules and settings that control the assembly process. You can create and save an unlimited number of libraries.

Book: The concept of a Pathagoras 'book' is quite simple. A 'book' simply is a pointer to the folder on your computer (or network) where your source clauses are stored. When Pathagoras makes a call on a book, it is simply reading the file names in the folder to which it points. On the other hand, recognizing the simplicity of 'what is a book' will lead to a much faster learning curve for Pathagoras. The terms 'book' and 'folder' can be used interchangeably.

- A '**folder**' is a standard, ordinary Windows folder containing Word documents. This is the most common type of book. See separate topic called [Folder of Terms](#)  69 for more additional information on creating and adding terms to folders.
- A '**glossary**' is a special, entirely optional, type of book. It is an advanced function of Pathagoras. You do not need to be familiar with this specialty 'book' to fully implement the program. But, in summary, a glossary is a single Word document that contains multiple clauses used to assembly documents. Instead of each clause residing inside separate documents within a folder, all terms reside in the same document. The individual terms are separated from each other by 'bookmarks.' See [Glossary](#)  70 for more information.

It is very important that you understand at this point that 'books' are *pointers* to locations, and nothing more. Books are *not* the actual text found at those location. By the same token, a 'library' is not some sort of 'super' folder which contains the actual text of its enclosed books. It is just a text file that contains the (up to 10) references to the books assigned to it.

Therefore, when we say that a library 'contains' books, we mean that a library contains the *pointers* to the various folders or glossaries reflected in the library. A 'library' is a simple 10 line text file that comprises your collection of 'book' pointers.

When you add a book, you simply are adding a pointer to that text file. When you delete a book, you are deleting the pointer from the library text file, not the target folder to which it points. Similarly, when you delete a Library, you merely are deleting the collection of pointers, not any substantive text.

This should explain how a library can 'contain' books, yet those books can reflect so many disparate physical locations.


Attempts to make these definition more 'complex' than "a book is a pointer to a folder" will lead to frustration. Pathagoras really is designed on a simple straightforward model. Please don't try to overthink it.

Clause: A 'clause' is the smallest component of text that you have designated for assembly. It can be anything – a single word, a phrase, a group of sentences, paragraphs, pages, pictures, charts, etc. A clause can even be a complete document.

- Clauses typically represent the component parts of a larger document. It includes 'boilerplate language,' but also includes text and images that will reflect the personality and personal data of the intended recipient.
- A good 'book' will contain many alternative versions of the same topic. You should make a concerted effort to have a wide selection of clauses from which to build documents. The more variations, the more 'powerful' the system, and the better the final document can be. With an appropriate collection of clauses, strategically organized in appropriate books and libraries, you can create an infinite variety of documents to meet all of your clients' or customers' needs. When you encounter a new variation of an existing clause, Pathagoras makes it easy to add it to your book.

Package: A group of documents typically produced for a customer as part of a single transaction. A real estate transaction 'package' would comprise a deed, deed of trust or mortgage, warranties and guarantees, and other closing documents. An estate planning 'package' would comprise a will, a trust, funding documents, a power of attorney, etc.

Variable: A 'variable' is a place holder for personal data. You should strategically place variables within your source clauses where you want that data to appear. Consequently, those variables will also appear in the first draft of any newly assembled document.

- Pathagoras prides itself on its use of plain text variables. Plain text variables are easy to create and insert into the source text. They are easy for operators and data entry clerks to understand when the final product needs to be personalized. See [Variables](#) 

Source clause: A 'source' clause is the 'original' version of the clause. It is the actual text stored within the books discussed above.

➔ Pathagoras stores source clauses in standard ".doc" or ".docx" (in Word2007/2010) files for easiest editing. See ['.doc' vs. '.dot'](#) ³⁰⁸

'Pathagorizing' (more accurately: 'neutering') is the process of preparing a document for use within the Pathagoras document assembly system. The process includes removing / redacting personal information from the original text and designating variables and optional text blocks which will be processed automatically by Pathagoras at document assembly time.

- The goal of 'Pathagorizing' is to make all clauses 'neutral' so that, after you bring it to the screen during a document assembly session, you can quickly and accurately personalize it for the specific client.
- The more neutral the document, the lower the chances will be that you will have a noun, pronoun or verb that is inappropriate in the context, or that you will accidentally leave the name of an earlier client's spouse as the beneficiary of the current client's Will!
- See separate section called [Pathagorizing text](#) ¹¹⁶ for more information.

Optional Text: Keep it or delete it text. [Fully discussed beginning at this link](#) ¹⁴¹.

Options Text: Multiple choice text. Select the text you want to keep, the other choices will be deleted. [Fully discussed at this link](#) ¹⁴¹.

Repeating text: Blocks of text that will be repeated 'X' number of times. Useful for listing actors (e.g., 'children,' 'shareholders,' etc. when the number of these actors is not initially known. [Fully discussed at this link](#) ³⁸².

Document Disassembly: The process of deconstructing a complete document into its component parts. The result is two (or twenty or two hundred) building blocks of text. The blocks in turn can be augmented by appropriate additional clauses offering variations and alternatives, and further augmented by bracketed variables and by strategically placed option, options and repeat text blocks. The ultimate goal is to be able to select from a large variety of available terms so that finely tailored initial draft can be created to quickly address the client's or customer's need.

Processing a document: 'Processing' refers to the handling of conditional text within the selected or assembled document. Processing typically precedes 'Personalizing' (discussed in the next paragraph. The end result of a 'processed document is a perfect 'rough draft.' Such draft contains the appropriate text, with all options and optional text block questions answered and repeat blocks handled. (You would want to 'process' the document first to rid the document of unnecessary and irrelevant text, including unnecessary variables.) Processing automatically occurs in Pathagoras when you call text to the editing screen via Pathagoras' two primary 'document assembly' tools: the (1) Clause Selection Screen (2) DropDown Lists. The end result of a 'processed document is a perfect 'rough draft' of the document.

Personalizing a document: The process of replacing bracketed variables with 'personal

text.' The Instant Database module (Alt-D) is invoked to display all document variables in the left column of that screen. The right side is used to provide the personal information.

Administrative text: Those portions of Options, Optional and Repeat text blocks contain the command term, the (optional) !groupname! of the block, the (optional) multiple choices labels that you may wish to present to the end users, and other processing instructions that are unique to the particular block being created. The end of administrative text is denoted by a properly placed closing asterisk.

Document Management:

Document management deals with quick and efficient storage and retrieval of documents.

Pathagoras' document *management* module is reflected in the PathSmart and SaveSmart modules. They shed the 'library/books' metaphor used in the document *assembly* module, and adopt the terms 'Profiles' and 'SmartPaths' to describe the groupings.

A '**SmartPath**' is a pointer to a specific folder on your computer or network. Each *SmartPath* is assigned a number and a nickname. The program can access the *SmartPath* by 'clicking' on the desired *SmartPath* or, mouselessly, by typing the *SmartPath* number to the screen and pressing <Alt-G>. (The module is called PathSmart, but each pointer is called a *SmartPath*.) See separate topic called [SmartPaths](#)⁵³⁶.

A '**profile**' is a collection of up to 10 'SmartPaths.' A profile groups the folders ('paths') that a particular user is likely to go for non-document assembly purposes. Since it reflects the usage pattern of a particular user, a profile is typically given the name of the current user or the name of an office section. See Also: [Settings](#)⁵⁴⁸

A '**SuperSmartPath**' is a regular SmartPath with one additional, but very powerful, feature. A *SuperSmartPath* allows you not only instant access to the folder to which it points, but to all sub-folders beneath it. See Also: [SuperSmartPaths](#)⁵³⁷.

1.2 Pathagoras in 1...2...3 steps

Because of Pathagoras plain text underpinnings, creating and using automated documents really is as simple as One . . . Two . . . Three.

1. Mark up a document by creating variables with simple square brackets. [Client Name] is a variable. So is [Customer Address]. Multiple choice variables are denoted simply by putting slashes between the choices. [chocolate/vanilla/Rocky Road], [he/she/it], [Joanne Doe/Robert Richards/Alex Attorney] are some examples.

Create optional (conditional text) using curly braces. { This is optional text. When encountered by Pathagoras at assembly time, the text will be automatically highlighted you will be asked whether you want to keep or discard the text. }

Save your Pathagorized document it into your library of forms.

2. When you want to prepare your saved document for your client, call up that document to the editing screen using one of the several assembly tools that Pathagoras provides. DropDown Lists are the easiest, but if you want to build a document from building blocks, the Clause

Selection screen is the better approach.

3. After the document is brought to the screen, Pathagoras automatically 'processes' the {Optional text} blocks first. (That way, if you have variables such as [Child 1], [Child 2] etc. in a document being prepared for one who has no children, those variables won't appear in variables list -- they will have been deleted when you answer the 'Do you want to keep this text?'" question.

So, with the processed 'rough draft' on your editing screen, press <Alt-D>. This displays the Instant Database screen. Click the Scan button and the plain text variables created in step **1.** above will be displayed in the left column. You simply provide personal values at the right. Click the Next button to replace variables with your clients' personal information throughout the document.

That's it! Pathagoras quickly replaces the variables throughout the document, ensuring consistency and saving hours of work! The values used for this document can be used in all other documents you create for the client.

Other features let you move from existing forms to automated forms quickly, easily and intuitively.

- If an existing form you wish to Pathagorize contains many of the same names (e.g., 'John Doe' appears in the title or caption of the document, in the body of the document and in the signature block):
 - highlight the first instance of one of those names
 - press Alt-V to activate the Create Variables wizard. Click 'Create Variables.'
 - provide a variable name (e.g., 'Client Name') to replace the existing name.
 - press 'Next'. All instances of that name are replaced with the new variable. (Even the upper and lower case attributes of the original text are preserved.)
- If you want a choice made in one section to carry forward to another:
 - augment the variable or optional text block with a 'group name'.
 - just inside the opening boundary mark, type an ! (exclamation mark), a group name (from 1 to 30 characters), and another !. E.g.:

On this day came [Client Name], [!sex!Testator/Testatrix] and declared that this is [!sex!his/her] Last Will
and that [!sex!he/signed] it as [!sex!his/her] free and voluntary act.

- All members of the same group will change in tandem. A group name is noted by a word or short phrase surrounded by exclamation marks.
- You can insert building blocks of text in two clicks from a DropDown List
- You can insert building blocks of text mouselessly by just typing the name of the document and pressing Alt-G).
- You can easily disassemble entire documents into those building blocks. Just highlight the text you want broken out and press Alt-G. Select the target folder and provide a name. That's it.

Hundreds of other features permeate this feature rich program. They are discussed at length in this Manual. (You can view a bullet pointed listing of the major program features -- over 200 of them -- at [this link](#).

At Pathagoras, we back everything up with extensive support seven days a week. And when you contact us, instead of getting an automated service, you're connected with a developer—who's there to answer any question you may have.

Pathagoras is available by outright purchase (you own the license forever; no annual renewals required) or by subscription.

1.3 Design & Work Flow Concepts

"Pathagoras is different from its competitors. Any similarity is purely coincidental."

Document assembly is, according to one definition, the process of creating documents from precedent clauses. When first introduced, it was thought by many to be the next revolution in legal technology. In actuality, it has been a bit of a bust.

Why would this be the case? After all, the envisioned cost and time savings that document assembly can bring to an office was unfathomable. But when reality sets in, it was found that most of those who attempt to implement a document assembly system have found the process of preparing the text, the interviews and the coding needed to make it all work to be overly complex and time consuming. Many a well intentioned soul began the process. Few finished it, and many of those who succeeded did so only with the (paid) help of consultants and experts specially trained in the particular program.

Enter Pathagoras. Our goal in developing Pathagoras was to create a dynamic, full-featured document assembly program that was simple, intuitive and actually enjoyable to use. We have designed Pathagoras so that every person in every office would want to use it, from 'Pathagorizing' a document (preparing it for use by the end user), to setting up the libraries and books of clauses (telling Pathagoras where the Will clauses, the Real Estate clauses, etc., are stored so they can be instantly displayed for selection) to personalizing the final document (including saving the variable to personal value match-ups so that data need only be entered once).

When we began to design Pathagoras, we began with no assumptions. This included an affirmative rejection of the concept that the work flow of existing programs were necessarily correct. We observed work flows in 'real' law offices. We heard the complaints and suggestions from active users of other document assembly programs. And we set to work and wrote the program around what we saw and what we heard. Here is a short list of some of the design requirements:

- One comment that we heard often was an insistence that the program run within and be a part of the underlying word processing system. Learning a new program just to make an existing one work faster was thought a bad idea. So we started writing the program using VBA, Microsoft® Word's macro language.
- There should be no dependency on hidden or 'out of document' fields. Creating fields (those grey blocks of text that changed values depending upon the embedded coding) was often confusing. Even if the document assembly program would create the fields for the end user, they were difficult to modify. It was not always obvious why a document assembled the way that it did because the coding for the document was stored 'somewhere else.'
- Creating menus was the most difficult part. Drawing meaningful screens needed to 'interview' the end user was tough. Writing the logic so that appropriate clauses would be selected based on the interview answers was even tougher.

- Were the tedious steps needed to create these interviews really necessary? "Why can't all potential clauses simply be displayed in front of us and we can pick and choose the one's we want?" The fancy stuff can come later. For now, "all I want is a quick way to get to my clauses."
- Many users wanted to be in total charge of what clauses would be inserted. They did not want a computer program telling them what clauses to use based on an interview. On the other hand, some offices use paralegals and other staff to create the base documents and the attorneys don't want the initial decisions made by a non-attorney. Use to power of computers to remember a specific pattern based on specific criteria.
- The program 'basics' should operate with little set up beyond installation. Significant parts of the program must be usable right 'out of the box.'
- The reason what a document works (or doesn't work) the way it does should be 'facial' to the source document. It should be a simple matter of looking at the source text, not ancillary files, to see why an assembled document looks the way that it does, and the source text should be easy to access so that any corrections can be quickly make and quickly tested.

We concluded that 'document assembly' is initially about getting a reasonable rough draft of a desired document(s) to the screen. What was 'reasonable' varied among users. Some have very low demands. Any text bearing any resemblance to the intended end product would do. Others wanted an almost 'perfect' first draft. (Those wanting the former far outweighed those wanting the latter. This clearly indicated that the complexities involved in getting a 'close-to-perfect' initial draft is what dissuaded even techno-savvy users from using the program more -- or at all.)

So here is work flow concept #1. *The quicker the end-user can get an acceptable version of the final desired text onto an editing screen, the happier that end-user will be.* And the happier the end user, the more likely that document assembly would actually be implemented. This is where we set our focus, and this is where Pathagoras shines.

We made the creation of document variables as simple as possible. Square brackets around placeholder words was to be the primary shape of a variable. And all text was to be 'plain text' (i.e., only regular keyboard characters would be used).

We practically eliminated the need to navigate up, down, across, further up, further down, your folder/network trees to find documents. But we did so while at the same time not requiring you to change a single thing about the way you currently organize your documents. And this is where Pathagoras begins to separate itself from the pack.

With Pathagoras, you avoid navigation by setting pointers to your existing folders, and assigning meaningful names to those pointers (typically the current folder name). Once set, you never manually navigate to that folder again. You use 'on-screen' cues and simple clicks of your mouse. The productivity savings with no-more-navigation is immeasurable.

Personalizing the assembled document to reflect the proper client or customer information was obviously important as well. And that is discussed next. But our users made it clear that just getting something on-screen to manipulate was the most important first step to document assembly nirvana.

Here are Pathagoras' two primary document assembly tools:

- **DropDown Lists:** This was mentioned in the introduction. Point up to 10 folders (or parent folders if you want to use the 'tree service') to Drop Down Lists and never navigate to them again. So if you have highly 'Pathagorized' documents or document which you have never prepared for automation, you can quickly bring an exact copy of any item in that folder to your screen by just clicking on the item. Nothing more.

- **Libraries and Books:** More clicks for a single document (so DDLs are preferred in that setting), but when you want to assemble multiple clauses or multiple documents, this is the way to go. An unlimited number of pointers to your most needed folders, arranged by topic (Estate Planning or Family Law) and specific subject within the library (Wills, Trusts, etc. or Divorce, Separation Agreements, etc.)

To call a document you wish to prepare for a client or customer, you would either call it from the DropDown List (2 clicks) or select the library then book then specific document(s) from the Clause Selection Screen. A rough draft of the document(s) you selected is instantly brought to the editing screen.

Once getting text to the screen became 'second nature', adding 'automation elements' to the source documents comes to the fore.

Work Flow Concept #2: Adding 'automation' elements to source documents must be easy and intuitive. With Pathagoras, automation is accomplished using plain text and remains intact on the face of the source document. Original documents need not be converted, renamed or moved.

Everything in and about Pathagoras is 'plain text' and 'facial'. Let's take these concepts one at a time:

Plain text:

- Variables are holding places for words you want to add when you are ready to personalize your document. Variables are created simply by surrounding words with square brackets. [Client Name] is a variable.
- Optional text blocks are 'keep it or leave it' blocks of text that may or may not remain in the final document. The decision is made at document assembly time. They are created by surrounding words, sentences or entire paragraphs with curly braces.
- Options text blocks is commonly known as 'multiple choice text'. They are text blocks from which you want to select at least one choice from several that are provided. Options text blocks likewise are words, sentences or entire paragraphs surrounded by curly braces. If you want to limit the choice to one, separated each choice by "/OR". If you want to allow the user to make more than one choice, use "/ANDOR" as the separator.

Just a note here. Some competitive programs also claim that their editing for automation is done in 'plain text.' There is a half-truth to their assertions. Initial automation markups can be done in plain text. The problem is that when you have finished editing and press the buttons to make it part of their systems, they convert your work into a new document, replacing your typing with their hidden fields and links. They sometimes move the document to a new location and almost always rename it, assigning a foreign extension and creating a tag-along document which contains additional instructions for processing the file.

None of that happens in Pathagoras. When you have finished editing your document, just save it. When you need to edit it again, you recall it from its same location. It starts as a Word document. All intermediate steps are done in Word and the final product is a Word document. We call this "All Word, all the time."

Facial text: The culmination of the above 'rant' is Pathagoras' concept of faciality. What you type is what you keep. No conversions. No fields. Document automation triggers are typed on the document's face and remain there until the document is automated. If your document does not process correctly, you can 'read' on the face of the document what went wrong. You do not have to consult the ancillary files (where ever they may be), edit the fields, convert and then reconvert. So 'Word-like' (here we are referring to Microsoft Word) is the process that when something goes wrong, you can simply press the Undo button on the assembled document. This lets you step backwards through the process

to see what Pathagoras did with the variables and options text you provided. This makes it very easy to see (albeit in reverse) what Pathagoras is doing with your markups so you can correct the original if the need be.

Once you have edited and saved your document, refer to Work Flow Concept #1 to access your document.

Here is BIG WARNING #1:

Never use an original document to serve as your base for a document assembly session.

So long as you abide by the above rule, you will not accidentally overwrite your source documents. (We are sure that it has never happened to you, but some people forget to save the edited documents under a new name. The original document is therefore lost. When you heed BIG WARNING #1, and follow the document assembly steps stated above, you minimize the chances of that occurring.

When a document is brought onto the screen using Pathagoras tools you will always be working on an exact *copy* of the original. How do you know it is a copy? Look for the 'name' of the document in the upper left hand corner of the screen. It will be named "Document1", "Document5", etc.

If you find yourself working on the original for other than source editing purposes, you should rethink your process, and implement the tools provided by the program. If you do nothing else in your early days with the program you should at least "[Create your First Library](#) ³²" and "[Shelve your First Book](#) ³⁵" and call up copies of documents from your various books. If you find yourself working primarily on complete documents or templates, consider our exceptionally popular [DropDown Lists](#) ³⁵⁶.

Adding New source text (Pathagoras Work Flow Concept #3).

Once you begin using Pathagoras, you will begin to muse, 'Gee, I sure wish I could easily and quickly add additional documents to the folders I assigned to my books or to my DropDown Lists.' (Well, maybe you won't state it quite that way, but the thought will nevertheless cross your mind.)

The process is (or course) simple, and no navigation is ever required.

Depending upon the source of the text you want to add, you have several choices.

Copying existing documents from 'other' folders:

- **To DropDown List folder:** If you are moving multiple files, copy files from the source folder by highlighting them, right clicking and "Copy". Then, click Display Folder from the appropriate Drop Down List. When the folder displays, right click and select "Paste." (Ctrl-C and Ctrl-V work as well.) If the text you want to move is 'on-screen,' simply highlight it (if you only want a portion). Drop down the appropriate DropDown List and select the "Save to Folder" element.
- **To Libraries and Books folder:** If you are moving multiple files, highlighting their names in the folder. (By folder, we mean simply the folder into which you navigated using regular Word/Windows navigation tools. You are simply locating files you want to add to a folder which you have identified to Pathagoras as a book.) Right click on your selection click "Copy". Close that menu. The selected documents are now in your 'clipboard.' Next, using the Document Assembly button, select the library and then the book into which you wish to add the new documents. Click "Display Folder" option from the expanded screen and click Next. A standard Word 'file open' dialog displays. After the folder displays, right click in any 'blank' area of the screen and click

"Paste." (Ctrl-C and Ctrl-V work as well to copy and paste the files.)

Moving 'on-screen' text into source folder:

- **To DropDown List folder:** Highlight the portion of text you want to add to the folder associated with your DropDown List. Drop down the appropriate List and select the "Save to Folder" element. Pathagoras will copy the highlighted text into a new document and save it with a name that you provide into the target folder.
- **To Libraries and Books folder:** Highlight the portion of text you want to add to the folder associated with you book. Select the appropriate Library and then Book and select the "Save to Folder" from the expanded screen. Click Next. Pathagoras will copy the highlighted text into a new document and save it with a name that you provide into the target folder.

(Of course, you can always move files into a target folder the 'old fashioned' way, and if you have several documents to move around, you can use standard file manipulation tools. But if you have 'free text' or a single document, the steps listed above cannot be beat for speed.)

Notice that all of the above is irrespective of the nature of the documents being moved.

Workflow Concept #3: There is no such thing as a 'Pathagoras document.' Pathagoras owns no document. There are only documents, and they carry the definition of documents according to Word and Windows, not Pathagoras. (Don't get us wrong. You can 'Pathagorize' a document by adding brackets around variables, and by adding Options and Optional text blocks, but the result is still Word document with the same name (unless you decide to change it) and extension (.doc or docx) as the one you began with.

Workflow Concept #4: Pathagoras goal is to may your document creation tasks easier, not different.

There is a lot more to the program, of course, than discussed above, but this should get you started with a solid understanding of Pathagoras' approach to document assembly.

Workflow Concept #5: It should be easy to add new text or to access for edit existing text to any document assembly system.

Marketing Concepts:

While we were at it, we decided to include our philosophy regarding marketing.

Marketing Concept #1: Demonstrate the program on the potential customers computer.

Marketing Concept #2: Show prices on the website.

Marketing Concept #3: Don't ask for life history to all the viewing of a sales video.

Marketing Concept #4: Don't nag. Every once in a while send a newsletter or ask how things are going, but don't nag.

Marketing Concept #5: Be as available before the sale as after it.

Marketing Concept #6: Be as available after the sale as before it.

Marketing Concept #7: Leave the user better off for having tried the program (or at least n worse off.) Example: Many customers 'coded' their variables like this: ***** and/or "_____". During the trial, replaced those 'variables' with much more meaningful variables like "[Date of Letter]", "[Client Name]" and "Child@1]". The customer who may decide not to buy doesn't have to undo the changes made, and actually has a better document in exchange for the trial. So your efforts at the trial are not wasted.

Marketing Concept #8: Offer features that are not dependent upon document coding. Allow the user to use the program when purchased, not just after the program has been fully configured. We can brag about two of these features: "[DropDown Lists](#)³⁵⁶" and the "[Names and Subjects Editor](#)⁴²⁸". These are spin off programs, potentially stand alone. By the same token, they are also at the heart and soul of Pathagoras.

If it is challenging to locate the source text to edit it, or to add more source text, those functions will not be done. So we had to make it easy and consistent with the way the typical user already functions.

When you want to create or edit source text, that too can be done without navigation. ('Source text' means the documents and other files that you want to actually store in the folders targeted by your books and DropDown List. It is from 'source text' that a copy is called when you take the above document assembly steps.)

Pathagoras' allows you quick access to your source text:

- **From DropDown Lists:** Drop down the list. Locate and click on the element called "Display Folder". A standard Word 'file open' dialog displays. Simply select the original file you wish to edit.
- **From Libraries and Books:** Click the document assembly button. Select the library and then book. Click the "Display Folder" option from the expanded screen and click Next. A standard Word 'file open' dialog displays. Simply select the original file you wish to edit.
- **Using existing Word tools:** Don't forget that you are in Word. Word maintains a Recently Used Files list. Use that list when you need to edit a file you just worked on. If you are editing, testing, reediting, retesting a document, this likely is the fastest way to access your work. And it is a perfectly acceptable (indeed, preferred) technique.

Pathagoras saves you time even if you don't 'Pathagorize' a single document. Even if you have inserted no brackets, no optional text, etc., into a single document, the fact that you can quickly call up a document onto the editing screen is a meaningful time saver. Some customers have bought the program for

the DropDown List feature alone.

When we demo, we demo when possible on your computer.

1.4 Processing vs. Personalizing

Document assembly using Pathagoras typically involved 2 steps:

- **Processing** -- Selecting or discarding the text blocks that will make up the initial draft of the document. Whether a clause appears in the initial draft of the assembled document is either based on affirmative selections of a specific clause by the user, or as a result of programmatic decisions made from values provided by the user. The 'result' of processing is a usable rough draft of the document which contains the desired substantive text. It still contains [variables], but it should contain only those variables that are pertinent to the desired document.
- **Personalizing** -- Assigning specific values to the variables that remain in the document after it has been 'processed.'

To become most efficient in using Pathagoras, it is imperative to understand the difference between 'processing' and 'personalizing' a document. . Both are easy.

Processing

'Document processing' includes those steps you take to arrive at the first 'good' draft of your document. Processing a document involves (1) assembling the document from source text (discussed below) and (2) answering Options/Optional/Repeats questions that appear in the source text. These Options/Optional and Repeats questions will automatically be presented when you have 'properly' assembled your document.

Many customers who are not yet familiar with the 'flow' of document assembly with Pathagoras will begin a document assembly project by recalling the source text onto the editing screen. That, of course, is the way they undertook a project in 'pre-Pathagoras' times. We need to 'break' you of that habit.

When you recall the original document, the text will just 'sit there' until you do something to cause a change. That is on purpose. Pathagoras thinks you simply want to edit the source text. Our take: *never* use the source text itself for active document assembly. Use the source text only to correct a typo or to add additional automation features.

It simply is too easy to accidentally overwrite the original when you start a document assembly session with a source document. So, Pathagoras requires that you 'assemble' your document to get the benefit of the program's automated features.

When you assemble your documents using the tools that Pathagoras provides (via DropDown List or Clause Selection screen), the process is active and automatic. Any <<*Options/Optional/Repeats*>> blocks present in the source text are 'processed' as you

desire. You end up with a 'perfect' initial draft, just what you were looking for.

- You know that you are looking at an assembled document because 'Document 1', 'Document 2', etc. shows at the top of the editing screen, indicating a new document -- one that you cannot accidentally overwrite the original if you press 'Save'.
- You know that you are looking at the 'source' document because its name shows at the top of the editing screen.

There are two primary methods to assemble documents using Pathagoras:

(1) Selecting documents or clauses from the **Clause Selection Screen**.

This is the 'classic' assembly process involving a click of the Document Assembly button, the selection of the proper library and then book from the library. When you click the Next button, the available clauses appear in the left panel. Select the desired documents/clauses and move them to the right ("Selected") panel. Another click of the Next button and the selected items are instantly assembled.

(2) **DropDown Lists**

If you have assigned a folder to one of your up-to-10 DropDown Lists, just click the list, point to the desired clause or document and click. The selected item is instantly brought into your editing screen.

When a document or clause is 'assembled' via any of the above techniques, Pathagoras is not done. Pathagoras searches for the existence of any <<*Options*>>, <<*Optional*>>, <<*Repeats*>> or <<text>> blocks that may have been brought onto the editing screen, and processes them.

Each assembly command calls in more text. That text add more variables to your document (or processing the text at hand might remove variables because the text is 'processed away' (e.g., answering 'no' to an <<*Optional*>> text block deletes the text block).

If the text processing involves responding to Repeats that contains variables within the scope of the <<*Repeat*>> block, variables will be added, each incremented by the designated number of repeats: [Buyer@1], [Buyer@2], etc.

If the <<*Options*>> or <<*Optional*>> blocks themselves bring in any more text, that new text is processed as well.

Processing continues unabated until Pathagoras has determined that there are no <<*Options*>>, <<*Optional*>>, <<*Repeats*>> or <<text>> blocks to process.

A third 'assembly' method does exist. It is the <Alt-G> hot-key. If you type the name of a document or clause onto your editing screen (and assuming that the document or clause is currently located in one of the "[Order of Search](#)"⁹⁵ locations), Pathagoras will instantly bring the selected item into your editing screen. However, we 'demote' this tool to this footnote, not because it is not exceptionally powerful and tremendously useful. However, we want to emphasize the Clause Selection Screen and the DropDown Lists as the primary and preferred document assembly tools. By all means, if you know how to use and like Alt-G, go for it. [Read more here.](#)⁵⁹⁰

Personalizing

When the document is fully 'processed' (i.e., you have a 'good' rough draft of the document with the clauses that you ultimately want), then 'personalizing' the document can begin. Your document contains not only the proper base language, but the proper variables (including variables incremented by the

Repeat processing).

If you are satisfied that you have a great base document, press <Alt-D> to bring up the Instant Database screen. Click the Scan button (or a Mask or an existing record) and Pathagoras will hunt for and display in the left-hand column all bracketed variables contained in the document. Provide replacement values at the right. Click the Next button and all variables are quickly replaced with their respective 'personal' text. (You will also be asked if you want to save the variable-to-values pairings to an Instant Database record. You typically should say 'Yes.')

What if I want to replace my variables first? What is the 'logic' behind 'process then personalize' order?:

Actually, there is a method to the madness. If document processing **starts** with <Alt-D>, the user is made to complete all variables that reside in the original 'unprocessed' document. However, as noted above, some variables could appear in the 'original' text that will be deleted when 'No' is the response to and <<*Optional*>> text question. Further, since new text (i.e., text that does not appear in the unprocessed document) that contains variables might be added to the document under construction by a call to an 'outside' document, you would have to run <Alt-D> again to pick up those new variables. So, to avoid the 'double' work, assemble the document first and let it be 'processed.' Then personalize it.

So when I call up one of my documents, it just sits there. Why is that?

Probably because you have displayed the original document as opposed to assembling the document. Remember what we said at the very top. Document assembly is performed by the methods discussed in the very first section. Document assembly should never be performed using an original document. Read, and heed, this warning:

Never use an original document as your base for a document assembly session.

So long as you abide by the above rule, you will not accidentally overwrite your source documents. (We are sure that it has never happened to you, but *some* people occasionally forget to save the edited documents under a new name. The original document is therefore lost. When you heed BIG WARNING #1, and follow the document assembly steps stated above, you minimize the chances of that occurring.

When a document is brought onto the screen using Pathagoras tools you will always be working on an exact *copy* of the original. How do you know it is a copy? Look for the 'name' of the document in the upper left hand corner of the screen. It will be named "Document1", "Document5", etc.

If you find yourself working on the original for other than source editing purposes, you should rethink your process, and implement the tools provided by the program. If you do nothing else in your early days with the program you should at least "[Create your First Library](#)³²" and "[Shelve your First Book](#)³⁵" and call up copies of documents from your various books. If you find yourself working primarily on complete documents or templates, consider our exceptionally popular [DropDown Lists](#)³⁵⁶.

So if you use Document Assembly Clause Selection Screen and DropDown Lists for your primary assembly, you won't get into 'trouble.' (Of course, if you are composing source text and want to process it for testing purposes, that is perfectly fine. Read [this section about testing](#)¹⁶⁰.)

1.5 Moving from basic to advanced

Pathagoras allows you to approach document assembly with a methodical, measured pace. It allows you to implement as much or as little of the program as you wish, as fast or as slowly as you wish. You can use the simplest of setups to get you started and then take advantage of the automation provided by Pathagoras to elevate the sophistication of your source text to any level you desire.

Unlike so many other programs out there, Pathagoras is not an 'all or nothing' type program. Start as simply as you like (or -- and we are not dissing anyone here -- as your skill level allows). Build on your successes on your own or using automation tools provided by Pathagoras.

Not a single step is wasted by starting 'simple.' You will not have to 'start over' if you decide to take a different approach. Pathagoras is replete with conversion tools so that you can experiment with different document assembly techniques. If you decide to try a different approach, use Pathagoras conversion tools to make the conversions for you.

As you begin creating your source text, don't try to 'eat the whole elephant' at the beginning. It can lead to frustration. Our suggestion is that you start 'Pathagorizing' your first form by creating 'simple' variables and 'simple' optional text blocks in a single relatively simple document. Play with that document to see and understand the action of Pathagoras. A 'simple' variable is a word surrounded by square brackets: [Client Name] is a simple variable. A simple optional text block is just a word, sentence, paragraph or group of paragraphs surrounded by curly braces. { This is optional text. }

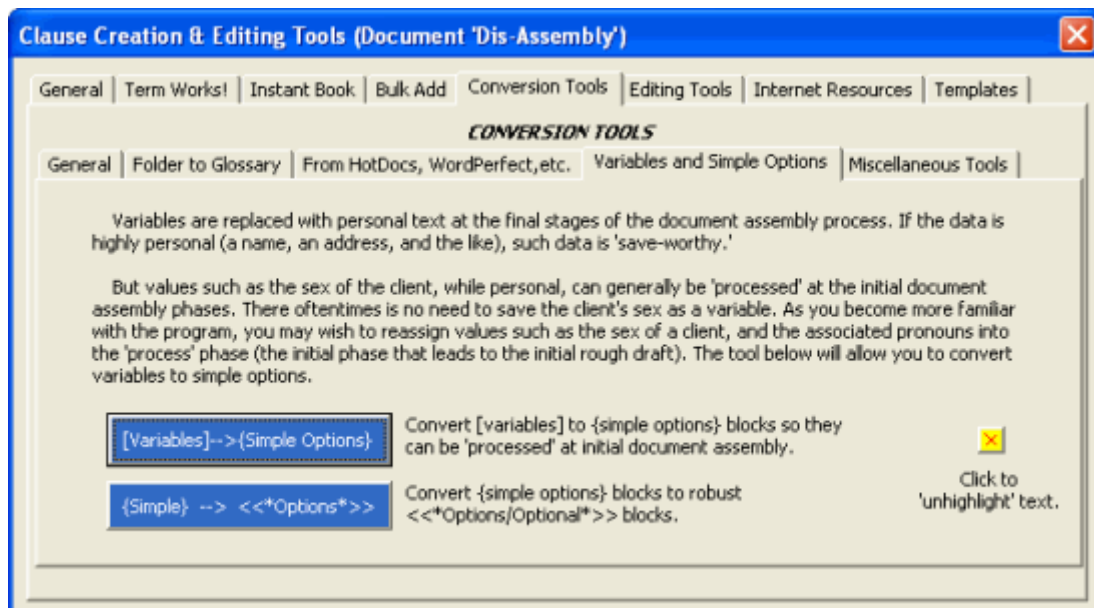
Once you appreciate the easy stuff, then move on to augment and advance the document to incorporate some of the more advanced features of Pathagoras.

Pathagoras conversion tools allow you to move your from 'basic' to 'advanced' in an automated fashion. These tools include:

- Convert variable to simple options. Most of the time, a variable should remain a variable. But sometimes the variables are more document structure. Examples, the sex of the client; the number of children. These items can be handled as variables (and for the beginner, they should), but the gender of the client and (perhaps more obviously) the presence or absence of children) are more 'document structure' in nature. Huge chunks of the document's rough draft would be eliminated if there are no children.
- Convert simple options to robust options/optional blocks. With these more robust blocks in place, you can present to the user more meaningful prompts that lead the user to the right choice. So instead of just showing the first 100 characters of the text of each of the options in the 'select' screen, you can present meaningful questions for the user to answer. Plus, with these robust blocks in place, you are just one step away from 'AskOptions.' Read more below.
- Convert 'robust blocks' to <<*AskPrompts">>. When Options/Optional and Repeats blocks are in place, with each augmented with a !groupname!, Pathagoras' wizard can collect all such blocks and place them at the 'top' of the document. When the document is processed, all questions are asked at the start and at one time. This avoids the repetitive ask one question/give one answer that the other processes use. This becomes the 'ultimate' in document assembly ease and speed.

Conversion tools can be found under the Wizards and Assistants section of the Pathagoras

Features list. When clicked you will be take to this screen:



Note the various tabs under the title line. You can convert from Folder to Glossary, from Glossary to folder; if you have HotDocs documents, you can convert the variables contained therein to simple variables recognized by Pathagoras. And more germane to the topic of this page, you can convert simple variables to simple options and simple options to robust options, all with just the click of a mouse.

Click on the appropriate button and follow the prompts that Pathagoras will provide.

1.6 File and Folder 'Pointers'

The 'plain text' operations that define Pathagoras are made possible because of its equally simple programming design. Pathagoras doesn't create a new subset of your documents, or create a database of your files requiring calls to retrieve or save a document to be intercepted by the program for initial processing. Pathagoras merely points to where your documents currently reside (or to the location where you save or move them using normal Word and Windows techniques).

An understanding of this incredibly simple 'pointer concept' is helpful if you wish to fully implement the program and integrate it into your practice.

➔ **A book is simply a pointer.** A Pathagoras book is a pointer to a folder. And a folder is just a standard Word/Windows folder, just like the kind you used before Pathagoras. It is a folder into which you can manually enter and from which you can manually retrieve clauses whether Pathagoras is in use or not. (It is nice to know this latter fact because if Pathagoras should ever 'crash', you are not stranded. Just navigate to the folder to retrieve the desired document.) When you assign a folder to a book shelf (let's say #8 represents the physical folder "c:\my documents\contract clauses"), Pathagoras lets you refer to the physical location by the number 8, but otherwise the folder is 100% intact.

By extension, a book is not the documents it points to (although to the end user, it definitely feels that way), but rather it is nothing more than a pointer to those documents. With that definition in mind, let's explore the positive consequences of this arrangement:

- You technically cannot add text to a 'book' (since the book is simply a pointer). You add text to a folder, just like you have always done in the past. It becomes part of the book simply because you added it to the folder. When Pathagoras rereads the folder's contents at the next document assembly session, the new document will be automatically added to the display.
- When you delete a document from the folder, the book is diminished simply by that action. When Pathagoras rereads the folder's contents at the next document assembly session, the deleted document won't display due to the mere fact that it is not present in the folder. You (as operator) don't need to perform any other task to keep your books in sync with your folders..
- When you delete a book, you are deleting just the pointer reference, not the physical folder. So you can add and remove books from shelves in your library with abandon.
- A book cannot exist independently from a library. (To be useful to you, these 'books/pointers' have to be stored somewhere, and the concept adopted by Pathagoras is that these books are collected within libraries.) Which leads us to the next main topic . . .

➔ **A library is a collection of pointers.** Following the same line as above, a library is not a large collection of files. The libraries refer to files, but are not the files themselves. (This differs from other programs where the files that make up the library are moved, encrypted or compressed into a single container. Rather, a library is no more than a collection of up to 10 books/pointers. Indeed, a library is actually only 10 lines in a simple text file. Each line in that text file will be either 'blank' (i.e., no book has been assigned to that line number) or a pointer to a folder in the style of "c:\my documents\contract clauses", etc.).

When you delete a book, you are not deleting the folder. You are deleting only the pointer which references the folder. Likewise, when you delete a library, you are merely deleting a collection of pointers, not any text in the actual folder to which the book points.

By way of illustration (and possibly by way of overkill, but it is important that we drive home this point): Let's say that you wrote the name of a folder on a piece of paper. You then took an eraser and scrubbed that folder name off the piece of paper. You have merely erased a bit of text, not the content of the folder that the words represented. If you want to physically delete or move a folder, you can do so, but must do so using standard Word/Windows techniques.

To see where a particular book is pointing, just hover your mouse over the book in the Libraries & Books screen. A small 'tip' box will appear telling you where it points.

➔ **DropDown Lists**, similarly, point to collection of files (typically documents, but actually can be anything) in a designated folder. The top-most and bottom-most elements of each List contain the information as to the whereabouts of that folder. When you click on a file name from the list, Pathagoras checks for the folder name and then quickly finds and inserts the appropriate file into your document.

➔ **Instant Database records** are individual text files typically saved on the user's computer. By default, Pathagoras can store them/find them because the initial IDB point is set to "C:\program files\Pathagoras\IDBs". But your Instant Database files can be stored anywhere and shared with anyone. To move them, just repoint the program (via the Instant Database settings screen) to the folder where you want your records stored. If you want others on the network to use those same files, just point them to the same location.

[Click here to learn how to set a new location](#)²⁵⁰ to which Pathagoras should point to find you Instant Database records.

Don't overthink this. Setting a location for your Instant Database records is merely 'pointing' Pathagoras to the location where those files are, or where you want them to be, stored. You are *not* activating a proprietary database where the records are encrypted, compressed and stored in a fashion that is inaccessible unless you have Pathagoras. To the contrary, your Instant Database files are stored in normal Word/Windows folder and readily accessible to any program.

See:

[Pointing/Repointing a Book](#)⁷⁸

[Pointing/Repointing to the Instant Database Records](#)²⁵⁰

[Pointing/Repointing to the MultiChoice *Lists*](#)⁴⁷⁸

[Pointing/Repointing to DropDown Lists](#)³⁶⁰

1.7 Advantages of Plain Text Document Assembly

There are several advantages to our plain text approach to document assembly. Here are a few:

- The learning curve much shallower. While there is syntax with which you have to become familiar, it is minimal and easy to learn. Pathagoras approach is undeniably simpler than the kind of field coding or setup required by others.
- Because it is 'all Word, all the time', you don't have to worry about [Client Name LIKE THIS] signals the other programs demand. Instead of the additional coding, just directly make variable look like how you want it to appear in the document. [CLIENT NAME] at the top (because you want the client's name spelled out in ALL CAPS) and [Client Name] in the body (because you want in Upper Lower case style) is just fine.
- The initial and the final 'coding' is facial. If your Pathagorized document does not process correctly, you can 'see' on the face of the document what went wrong. You do not have to consult the remote tables or ancillary files (where ever they may be) to edit the fields, convert and then reconvert the source text.
- Because it is 'all Word, all the time', if something goes wrong, you can simply press the Undo button on the assembled document. This lets you step backwards through what Pathagoras did in assembling the document. This makes it very easy to uncover (albeit in reverse) what happened to your markups so you can correct the original if needed.
- 'Plain text' means that you can 'Pathagorize' your source documents using a computer on which Pathagoras has not been installed! (Of course, to *test* your work, you will need a computer on which Pathagoras has been installed.)
- Any commercial document, document that you find on line, or document that you pick up at a continuing professional education class, that presents variables within variables is automatically a Pathagorized document.
- Perhaps most importantly, if you happen not to have the identical job in the office one year (or 10 years) from now, your successor has a fighting chance of figuring out what you have done.

All references to "Word" refer to Microsoft® Word, a product of the Microsoft® Corporation.

1.8 Document Assembly vs. Document Management

Pathagoras contains both Document Assembly and Document Management modules. As used by Pathagoras, these two modules are distinct, but integrally related. The access to the two functions is controlled by different screens, but the underlying programming code frequently overlap. It is important to understand, and distinguish between, these two concepts.

- The two modules are not mutually exclusive. They are not polar opposites. Rather, they are distinct components of the same system, and are designed complement and support each other quite nicely.
- Typically, a user will create and personalize a new document by using the Document Assembly module. After the initial editing is done, the user will store the new document, and recall it for later editing or printing, using PathSmart/SaveSmart (document management) tools.
- A folder mapped to a PathSmart 'profile' (collection of SmartPaths) can be the same folder as a book mapped in the Document Assembly 'library' (collection of Books). For example, the path to 'Office Forms' might be both a Book in a library and a SmartPath in profile. But they are used for different purposes.
- However, a folder containing personalized client documents (mailed letters, completed contracts) would *never* be part of a Document Assembly library. Why? Because personal documents would never be directly used to create future documents. (At least we hope that you never directly use a form created for one person as the source of one you intend to create for another. Doing so can lead to embarrassing, if not disastrous, consequences. 'Pathagorize' it and save it to a forms book.)
- A document retrieved from a SmartPath will typically be the original document. When you use the document management tools, Pathagoras presumes you want to edit the original text (modify content, correct errors, etc.).
- A document retrieved from a Book will typically be a copy of the original. When you use the document assembly module, Pathagoras presumes you are creating new content using original sources, and protect you from accidentally overwriting the original.
- Folders containing personal client letters, contracts awaiting execution, pleadings for a specific case and other personalized documents are precisely the kind that would be mapped within a PathSmart profile. Folders containing form documents, clauses, templates, glossaries, etc, that you view as 'source' text for creating new content are what would be mapped to a Book.
- There are other differences among the modules, some subtle, some more blatant. Only actual use of the program will bring into sharp focus the differences between them. Just take comfort in knowing that both systems work seamlessly together.

The following further illustrates the differences between document assembly and document management. We will stick with the libraries and books metaphor described in previous sections.

- Let's assume that you have been assigned to research and prepare a report on flying buttresses for an architecture project. To accomplish your task, you would go to the appropriate architecture library to find the information you need. You would look in the various books there. You would extract text and copy other source materials for the final project. And you would assemble your research into what ultimately becomes the first draft of your final report.
- After you have completed the report, you would not return to the library to store your composition. Rather, you would store your work in a location appropriate to a specific 'personal' project. You would also (mentally) use a methodology for storing client/customer projects that is distinctly different from the methodology used by the library from which you gathered the source material.

In all likelihood, you would place your final research product inside the customer's folder or a client's file, organized by customer/client name. (If it is a general office project, you might place it in a general purpose filing cabinet.)

But one place you *don't* store it is back in the library.

The same concepts pertain to the primary systems that make up Pathagoras. The document *assembly* system provides the tools needed to create documents from source clauses. But once created, the document *management* system provides the tools by which you can store and easily retrieve the completed documents from personalized storage containers.

'Nuff said?

1.9 Beyond 'Major' Projects

Pathagoras is a multi-dimensional program. It can easily be used in every aspect of office operations. Pathagoras can assemble complex trusts, lengthy bids and detailed requests for proposals. It can be used to put together highly personalized pleadings and contracts.

But Pathagoras' should be used with everyday cover letters and memos, too. Its simplicity and speed lends itself to such use.

Think broadly and creatively when it comes to implementing Pathagoras. It can save you an incredible amount of production time. There is no reason why every form and letter in your office should not be Pathagorized.

- Call up your frequently used documents and forms and put [brackets] around every potential variable.
- Assign folders containing frequently used forms and letter to a [DropDown List](#)³⁵⁶ and 'point and click' instead of navigating.
- Use the [Instant Database](#)¹⁸⁶ module to quickly identify and replace each variable.
- [<Alt-G>](#)⁹⁴ can call in not only the ultra-sophisticated blocks of text into your work screen, but also simple signature blocks, letterheads and other text frequently used in everyday office operations.

Unlike other programs that require extensive setups that would dissuade their use on your 'simple' forms, Pathagoras is the opposite. Its plain-text roots make 'Pathagorizing' your entire office quick and easy. A little bit everyday is all that it takes.

1.10 Help Connections

Other Built-in or On-line Help Systems


- ➔ Check out our [FAQs](#) found on our site. Read what others are asking; pick up so good hints and tips; read what others have posted as 'challenges' and our (or others') solutions to those challenges. We think that the FAQs are as important a resource for the efficient use of Pathagoras as is this Manual.
- ➔ Every overlay screen shown by Pathagoras contains at least one box containing a question mark. ("?") or Globe. These symbols are fully integrated with the Pathagoras Help System. Whenever you click one of the '?'s or Globes on a screen, you will be taken to the appropriate topic in this system.
- ➔ A downloadable, printable PDF version of this Help System is available. Click here for the details.
- ➔ Two shorter guides designed to help the newcomer become familiar with the scope and features of the program are available. Check out:
 - ['Beginner's Guide to Pathagoras'](#) This Guide contains illustrated, step-by-step instructions for the program's basic features.
 - [The 7-day Plan](#): Spend 15 to 20 minutes each day leaning the program basics. Specifically written for those who "don't have enough time" to learn another program. The most popular of our beginner's guides.
- ➔ An 'in-line' help system can be activated by clicking the "How Do I . . .?" element found either in the Main Menu or at the bottom of the Pathagoras dropdown features list. This context sensitive help tool provides short tips and 'how to' steps to accomplish a given function.

1.11 Upgrading Pathagoras

Pathagoras is always changing. It is upgraded on a fairly regular basis as users offer suggestions to improve current features or to add new ones. It is this reason why Pathagoras is distributed only over the Internet.

To upgrade Pathagoras,

1. Display the Utilities/Settings screen from the Pathagoras dropdown features list.
2. Select the Upgrade tab. (You can check for the latest release version number by clicking the 'Check for Upgrade' button. Compare it to the information on the Upgrade tab to see if an upgrade is indicated.)
3. Click the button that reads "Download complete program from the Internet" and follow the prompts.

 Upgrading Pathagoras is a simple, painless process:

- Pathagoras preserves all libraries, books, settings, Instant Database records, etc. You will lose nothing when you upgrade.
- You do not have to uninstall a previous version of Pathagoras to upgrade.
- The transition from a demo version to a licensed version is seamless..

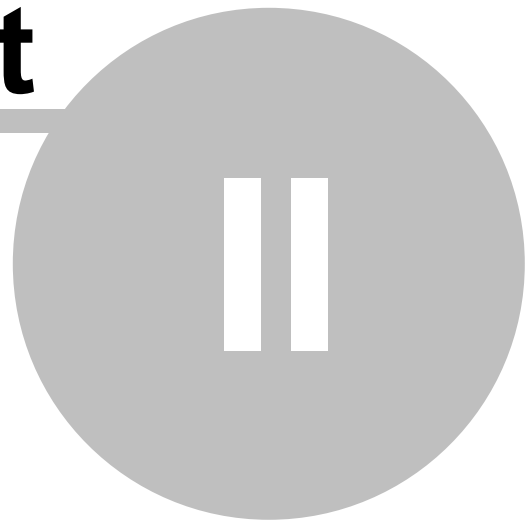
i **Network users:** follow the special [upgrade instructions](#)⁵⁶⁴ to most efficiently upgrade Pathagoras in a network setting.

→ You are entitled to the latest version of Pathagoras while you are within the period your initial Annual Support Agreement or subsequent renewals.
If you are not sure if you are 'current' in your Support subscription, see [this section](#)⁶⁸⁷ of the Manual or [contact us](#)⁶⁹⁰.

The Pathagoras System

Getting Started

Part



2 Getting Started

Pathagoras requires practically no preparation in order for you to use its basic features.

If your forms are fairly well organized (for example, you have at least one folder of forms from which you typically draw basic documents), you can take advantage of two of Pathagoras major functions immediately upon installing the program. With no further preparation, you can:

- [place that folder onto a bookshelf](#)³⁵ in the Document Assembly Library and begin preparing document for clients, customers or patients.
- assign that folder to a [DropDown List](#)³⁵⁶. With such a list, you can point and click any item in the folder into a current or a new document with no navigation.

If you do not have any forms segregated into forms folders, you should work on doing that fairly early in the process.

This section is a slightly shorter version of the separate "[Beginner's Guide](#)" written especially for those just getting started with Pathagoras. If you need a bit more detail in setting-up the foundation for a document assembly system, we invite you to print and peruse that Guide.

2.1 Download and Install

Downloading Pathagoras

Pathagoras is distributed exclusively over the Internet. There is no 'boxed' version of the program.

Retail Version: When you purchase a retail license from the Pathagoras website, you will be directed to a site from which to download the full Pathagoras program. You will also be provided a serial number with your purchase receipt or confirmation by which you can unlock the program.

Demo version: Pathagoras also offers a free 90-day trial version. Download the Pathagoras Demo from the Pathagoras website at www.pathagoras.com.

You should save the downloaded file (called Pathagoras Setup.exe) to a location on your computer where you can readily find it. We recommend the desktop, at least until you have installed Pathagoras. Once installed, you can erase the Setup file.

Installing Pathagoras

After downloading Pathagoras from the appropriate source, double-click on the Pathagoras Setup.exe file and follow the prompts. All files required by the program will be moved to their proper locations.

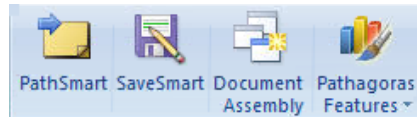
After you have installed Pathagoras, it will automatically open and automatically close alongside Word.

Access to Pathagoras' features is done through the new toolbar that now resides in the menu area of your screen:

If you are using Word 2000 through 2003 the Pathagoras Toolbar will look like this:



If you are using Word 2007 or later, the Pathagoras Toolbar sits behind the Pathagoras tab and looks like this:



Setting up Pathagoras

Pathagoras requires very little preparation in order to use its basic features. See the section entitled [Getting Started](#)³⁰ for help on 'jumping right in.'

Running Pathagoras

Actually, you don't 'run' or 'start' Pathagoras. Being a Word add-on, Pathagoras 'runs' whenever you run Word, and closes whenever you close Word. You don't have to do anything special beyond the installation discussed above to cause Pathagoras to run.

But what about the Pathagoras icon on my desktop?

That's just the installer. It's called "Pathagoras Setup.exe". Once Pathagoras has been installed, you can delete it that program.

See Also:

[The Pathagoras 4-element Toolbar](#)⁶⁵⁴

Installation Problems

2.2 Prepare Folder(s) for Document Assembly

To begin using Pathagoras, you should have, or should create, at least one folder which contains actual forms that you intend to use for document assembly purposes. (A folder which contains documents that you will be converting into forms will also work).

If you do not have a folder in mind, then create one. Navigate to the location where you want the folder to be. Its location is not critical at the present time. You might want to call this new folder "Pathagoras Forms" just to identify it as 'special' while you go through the initial exercises.

Whether you are using an existing folder or have created a new one, pour 'lots' of forms or documents into it. Put in whatever relates to the subject matter. (Don't worry about getting all documents and forms into folder right now. You can always add more later.) Populating the folder with 'lots' of documents is more important at this stage than even having 'well-Pathagorized' documents.

➔ **Note:** Creating folders and sub-folder, and moving documents into them, are standard Windows and Word activities. Use the "copy, cut and paste" and "drag and drop" techniques that you already know to move documents into the folder.

Pathagoras does best with lots of documents in a single folder. It is actually better (and easier) to have dozens of documents in one or 2 folders than to have one or two documents in dozens of folders.

2.3 Creating a New Library

The Demo Settings, Profiles and Libraries that ship with Pathagoras are good tools to help you to learn the program's basic features. However, before you begin to make changes or additions that you wish to be permanent, you must first create a new library. This is because every time you download an upgrade to Pathagoras, the DocAssemDemo systems are overwritten. Here are the steps:

1. Click the Document Assembly icon.



Figure 1. The Pathagoras Toolbar.

The arrow points to Document Assembly icon.

(Word2007/2010 users: Click the Pathagoras tab to reveal the toolbar.)

The Libraries & Books screen (figure2) appears. The default 'DocAssemDemo' library is on display.

2. Drop down the list of available Libraries. Click the <<Add New Library>> entry.

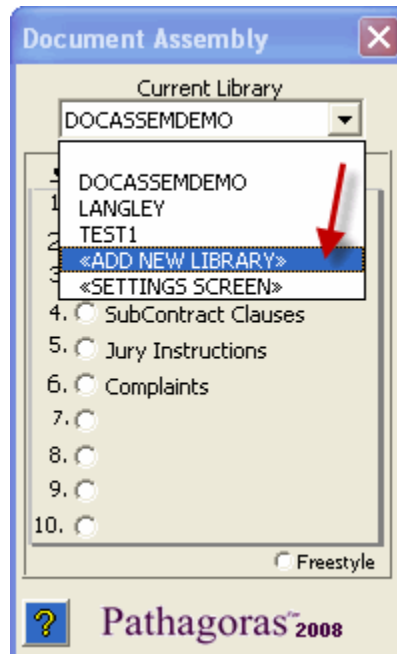


Figure 2. Initial Document Assembly Screen

The following screen will appear:

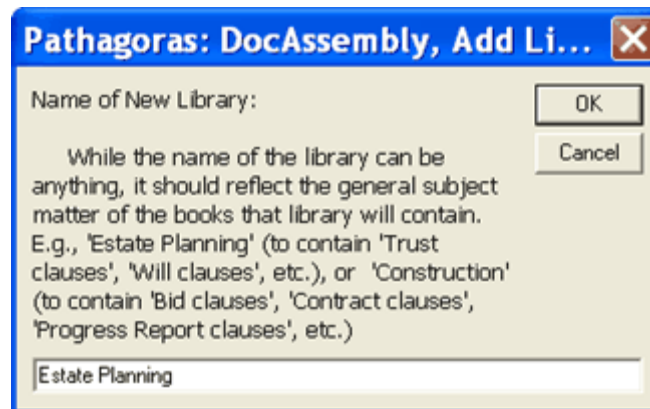


Figure 3. Document Assembly Settings screen

3. Provide a name for your new library in the screen. Click OK.

After you click OK, the Document Assembly 'Settings' screen will appear, showing the previous Default Library's settings (Figure 4). It will then lay over that a screen (Figure 5) asking how much, if any, of the previous Library's settings you wish to keep.

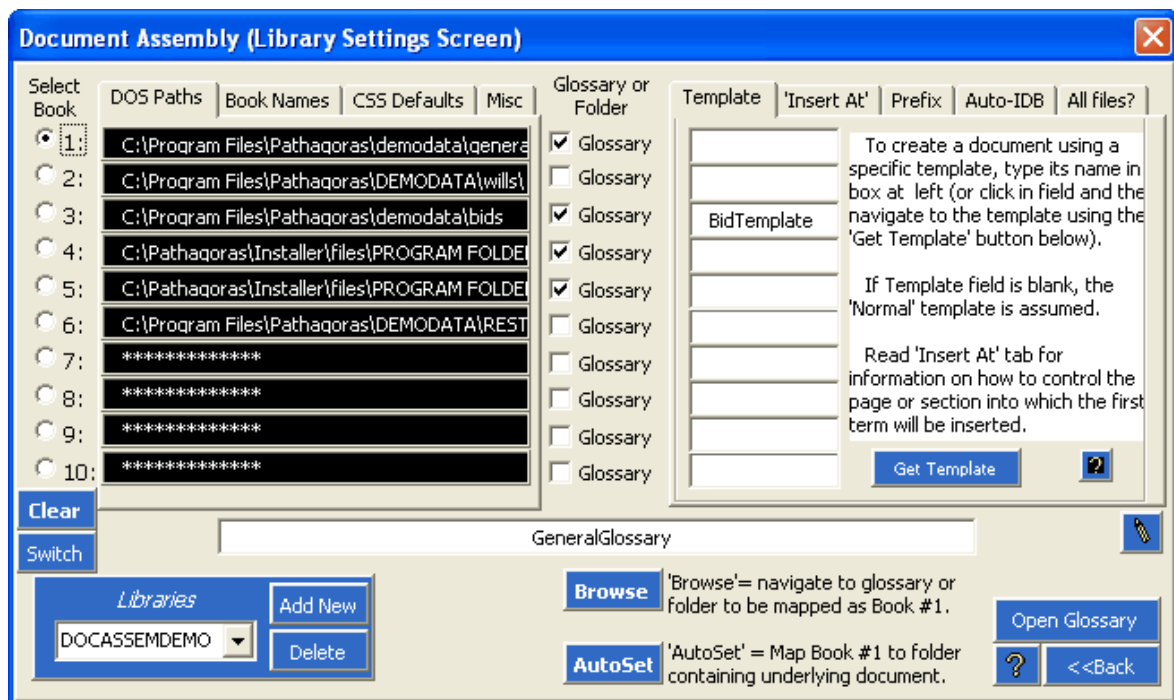


Figure 4. Document Assembly Settings screen

4. When Figure 5 displays, make a choice. Click the top button if you want to preserve the current setup. (If you preserve the current setup, you can change any setting later.) If you want none of them, click the second entry and your new library will be empty. For this exercise, let's start 'from a blank sheet.' Click the second button.

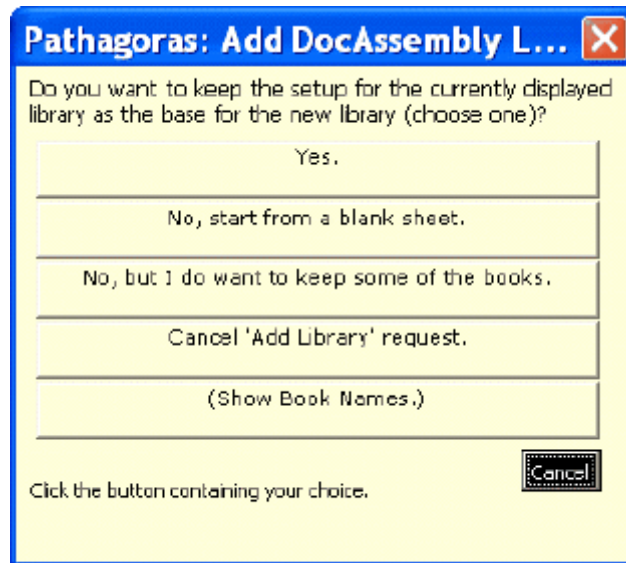


Figure 5. Let Pathagoras know which elements of the 'old' Library's Settings (if any) you want to carry over to your new Library

5. The screen shown in Figure 5 will disappear, leaving you with an empty library displaying in the Library Settings screen. You can put books on the shelves by clicking the Browse or AutoSet buttons (near the bottom). But we are going to add books a different way for this exercise. "X" out (or otherwise close) the Library Settings screen, and proceed to the next exercise, "Shelving Your First Book."

You now have (at least) two libraries: (1) the original DocAssemDemo library and (2) the one you just created (and any other ones you may have earlier created) . The last one you created will be the new 'default' library.

You can see all libraries, and choose among them, by clicking the drop-down list under the words "Current Library" in the Libraries & Books screen (Figure 2 above). The last one you select will become the new 'default' library.

Alternative method: You can add a new Library directly from the Settings screen. Click the Settings button in the lower right corner of the Document Assembly screen. Then, when the Settings screen appears, click the Add New button and follow the prompts.

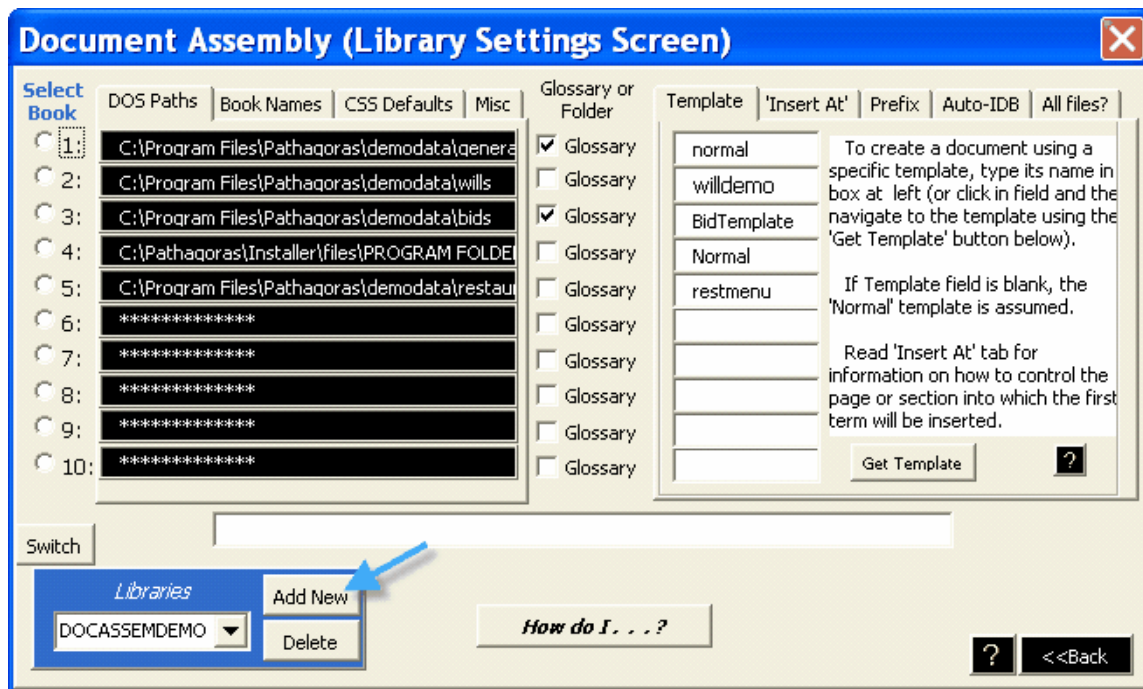


Figure 6. Instead of adding a new library from the initial Document Assembly ('Libraries & Books') screen, you can activate the Setting screen and click the 'Add New' button there. The remaining steps are identical.

➔ When you upgrade Pathagoras, the installer will overwrite the 'Demo' profile and the 'DocAssemDemo' library. (No other profiles or libraries are affected.) If you made changes to either of these sample setups, they will be lost. Be sure to copy them over to a new profile/library using the above instructions.

2.4 Shelving Your First Book

In Pathagoras, the term 'shelving' just means 'pointing' or 'linking.' In this exercise, we are going to point one of the blank 'shelf' in the current document assembly library to a folder of your choice. To reiterate, a book is nothing more than a pointer to where Pathagoras will look for certain forms and documents. Please keep in mind that the book is not the documents themselves.

To place a book on a shelf in your new library:

1. Click the Document Assembly icon as shown below.



Figure 1. The Pathagoras Toolbar.
The arrow points to Document Assembly icon.

The Libraries & Books screen (with the DocAssemDemo library on display) appears.

2. Click on any blank line. A new item "Quick Set" appears toward the right side of the screen. (See Figure 2, below.) Click it.

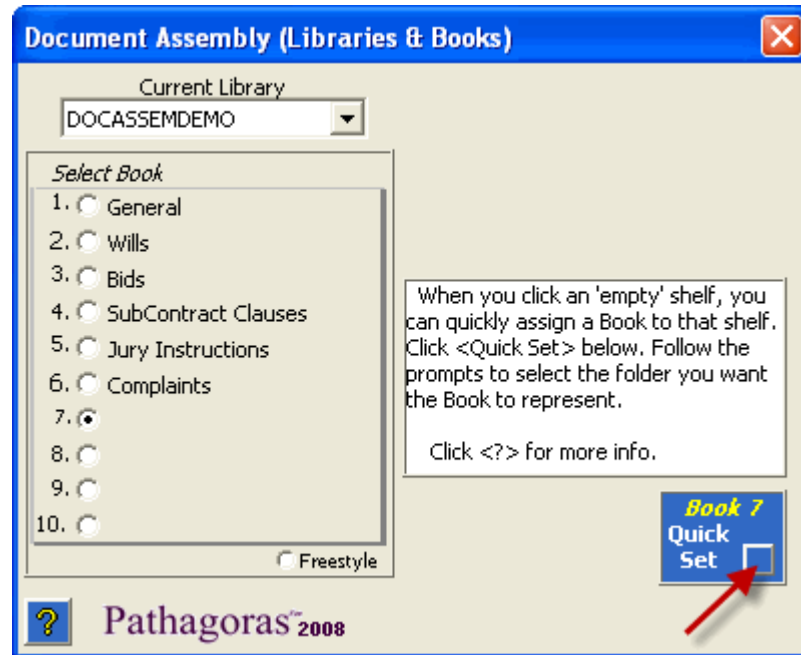


Figure 2.

When an empty shelf in the library is selected, the Quick Set button displays.

3. A new screen appears. (Figure 3.) It offers you choices as to what type of documents you want this book to contain. Pathagoras allows several options, but for now, click the first entry. (It's called "all .DOCUMENTS (*.doc files) in a selected folder").

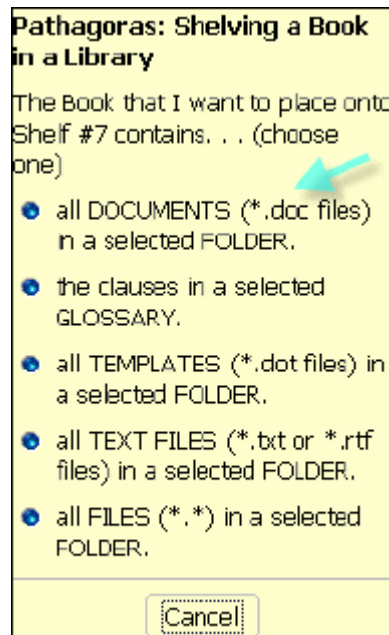
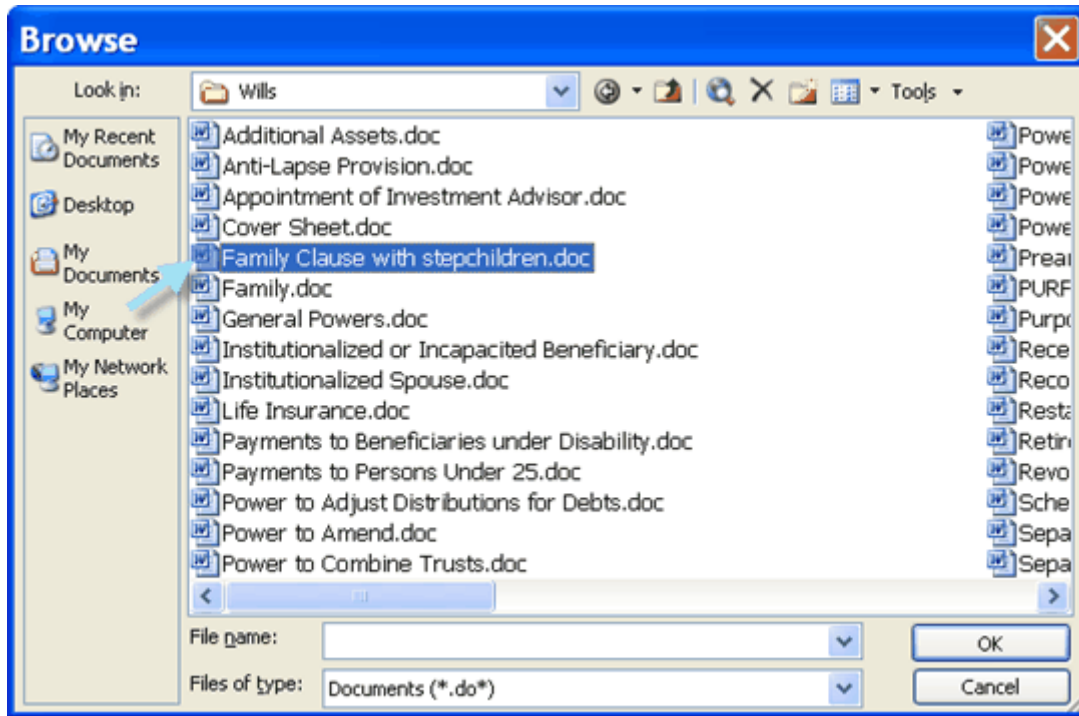


Figure 3.

4. Navigate to the folder that contains the documents or forms you wish to place onto the selected bookshelf.
5. Select one document from the folder and then click OK.

➔ **NOTE:** *It absolutely does not matter which document you select. The selection is only to confirm that you have finished navigating, that you are inside of the desired folder, and so that the OK button can become 'hot.'*



Navigate to the folder that contains the documents.
Select any document in the folder. Then press OK.

6. You will next be asked to provide a name for the book. The name of the folder will be proposed, but any name you choose is fine. It should reflect the nature of the book.
7. Press the remaining “OK” buttons. Accept any remaining defaults. If you are presented a screen that you do not now know how to complete, just leave it blank. It's fine! The ‘right’ answers are just not critical at this stage. You will learn more about those features later.
8. The book is now ‘shelved’ in your new library. Pathagoras will display a “Success” screen reflecting that fact.
9. You can place other books onto the library shelves by repeating the above steps.

2.5 'Pathagorizing' Your First Form

It should take you less than 10 minutes from start to finish to complete this exercise. If you are creating a lot of variables, stop after about 5 minutes so that you can test your work. Finish it up later.

1. Open a document that you want to 'Pathagorize' (neuter). This document should be from the folder that your first book represents. (See previous section.)
2. Study the document. Look at all of the places where 'personal' or customer- or client-specific data presently exists. We are talking about names, addresses, dates,

quantities, colors, etc. Anything that is likely to be different from one completed form to the next.

3. At the beginning of each of those 'personal places' type a "[" (a bracket, no quotes, tho'). At the end of each 'personal place' type a "]" (again no quotes).
4. Replace the stuff in between the two facing brackets with generic sounding, not 'real people,' terms. E.g., "[Customer Name]"; "[Names of Children]"; "[product description]"; "[shipping date]", etc.
5. So, after completing the above steps, "Jonathan E. Doeberg" might become "[Client Name]" and "4,342,654 widgets" might become "[quantity] [item ordered]". If "Mr. Jonathan E. Doeberg" appears in the address line, make the address line something like this: [Title] [Client Name].
6. Create an 'optional text' block. This is 'take or leave it' text. When encountered, Pathagoras will ask if the text should remain in the document.
 - Choose (or create) a sentence or paragraph that you want to be 'optional.'
 - At the beginning of the text, type "{" (no quotes).
 - At the end of the optional text block, type "}" (again, no quotes).

Example:

{ Please call me as soon so that we can discuss the various options that I have outlined in this letter. }

7. Save this new document. If the name of the document bears the name of the original client, do a 'SaveAs' and give the Pathagorized document an appropriate 'form' name.

You have now placed your first book onto a shelf in your library. This makes all documents in that folder available for document assembly. You have also 'Pathagorized' your first form. Repeat as appropriate to build your document assembly foundation.

The next sections of this Guide will take you through the actual processes of document assembly. When asked to select a particular book or navigate to a folder, you can use the books, folders and forms created in this section.

2.6 Installation Checklist

- ☐ **Install 'full license' version of Pathagoras to the new computer.**

With just the install, you can immediately begin to create variables, scan the 'Pathagorized' and produce personalized text. It really is that simple.

Muti-user licenses:

- ☐ **Check pointers to your Instant Database records. If you want to share Instant Database records with others on the network, you should create a folder on a computer to which all users can point. Then, point the IDB path on your computer to that location and point the IDB path on all other computers to that same location.**
- ☐ **Check pointers to your MultiChoice *Lists* file. (By default, but not by necessity,**

the MultiChoice *Lists* file moves in tandem with your Instant Database records. But if you elect to store them in separate locations, make sure that all users have pointed their MultiChoice *Lists* to the same folder.

Additional Network tools:

Keep in mind the following: Pathagoras can span your network with no setup beyond the initial installation. It is not necessary to take any of the below steps in order for Pathagoras to use and assemble document across the network. But if you, as administrator, wish to share Library and Profile setups that you have created on one computer with the other computers on the network, then you should perform the following steps.

- ☐ **Create a Common Profiles Path.**
 - ☐ **Check pointers to your various books in your Libraries.**

 - ☐ **Check pointers within MultChoice *Lists* files to external databases.**
- With all settings checked and confirmed, you should be ready to roll.**

The Pathagoras System

Document Assembly

Part



3 Document Assembly

'*Document assembly*' is the process by which an operator creates an entire document from a variety of component parts and then personalizes that document to meet the needs of the intended recipient.

Included within the scope of the term 'document assembly' are the means by which the source clauses are:

- created
- neutered, and
- assembled.

Pathagoras is a complete document assembly system. As such, it comprises many different facets. The primary objective of the program is to make each of those facets as simple and intuitive as possible.

Pathagoras provides two distinct ways to 'assemble' documents.

I. The "classic" method is dubbed '*paragraph assembly*'. The setup for paragraph assembly is simple:

Create folder to house a collection of files related to a particular subject. For example, place a collection of clauses used to build your wills in a folder called "Wills". Put a collection of clauses used to create real estate contracts in a folder called "Real Estate Contracts". (The name of the folder is not critical. The content is what matters. If you already have a collection of clauses reasonably broken down by subject, you do not need to create new content to implement Pathagoras. What you have, and where you have it is likely just fine.)

We call the folder in which you have your clauses 'book'. To repeat, the content of the book need not be new content. Pathagoras works well with anything you have.

Each file in the book would (typically) contain a paragraph (or two, or several) of a complete document. These text blocks, when combined with other documents during the paragraph assembly process, would create a 'perfect' initial draft for a specific project.

On command, Pathagoras will display all of these snippets onto a 'clause selection screen.' Select one, several or all of them. Once you have selected the appropriate clauses, press the Next button. The selected clauses are quickly cobbled together into a complete document.

Paragraph assembly is discussed in much greater detail on the [following page](#)⁴³.

II. A second technique is 'assembly from templates' (or, imperfectly, 'template assembly').

A 'template' is simply a generic form document that contains every likely clause and variation needed to produce a final document.

'Template assembly' doesn't offer quite as many permutations of possible documents as does paragraph assembly, at least not without greater effort. Plus, when an address, statutory reference, firm or attorney name, or other piece of text found in all templates changes, you are forced to make the changes in all of the templates.

However, for beginners to the document assembly administration, template assembly is quite a bit easier to implement. (The document '*dis-assembly*' aspects of system preparation -- when the individual clauses and paragraphs needed for paragraph assembly are created -- are postponed until a time when the office is 'ready' for that level of assembly power.)

Assembly from templates is discussed under ['Technique 2'](#)⁴⁵ below.

For a more detailed comparison between 'Paragraph Assembly' using clause snippets and 'Assembly using Templates', [click here](#)⁶²².

➔ An Overriding 'Pathagorean' Rule:

Never use an original document as your 'base' for a document assembly session. So long as you abide by this rule, you will not accidentally overwrite the original by forgetting to save the edited document under a new name.

Of course, there is one exception to this rule. When you are *intentionally* editing the original, with the goal of improving the source text itself, correcting spelling errors, etc., you must open and work on the original document. But NEVER NEVER NEVER call up the original document with the idea of editing it into a final document for a specific client or customer.

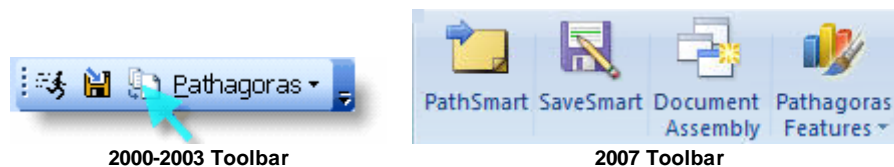
When set-up following the guidelines in this Manual, Pathagoras automatically implements this rule for you. When a document is brought onto the screen using Pathagoras tools you will always be working on an exact *copy* of the original. How do you know it is a copy? Look for the 'name' of the document in the upper left hand corner of the screen. It will be named "Document1", "Document5", etc.

If you find yourself working on the original for other than source editing purposes, you should rethink your process, and implement the tools provided by the program. If you do nothing else in your early days with the program you should at least "[Create your First Library](#)"³² and "[Shelve your First Book](#)"³⁵ and call up copies of documents from your various books. If you find yourself working primarily on complete documents or templates, consider our exceptionally popular [DropDown Lists](#)³⁵⁶.

3.1 The Process, Technique 1

Paragraph Assembly:

1. Click the Document Assembly icon:



The Pathagoras Toolbars.

The Libraries & Books screen will appear.

2. Select the book containing the clauses you wish to assemble.
3. Click the [Next>](#) button. The Clause Selection Screen displays, showing all of the documents in your new book.

Note: If you are performing this exercise and have selected one of the demo books, you might see rather cryptic names such as 'wil100', 'wil120', etc. in the display. To view the more descriptive 'subjects' associated with these terms, click the 'Subjects' button at the right side of the display.

4. Click on one, several or all of the items you see in the left panel.
5. Click the Add>> button to move your selection from the left column to the right. (You can also double click on the selection to accomplish the transfer.) Repeat until you are happy with your selections.
6. Click the [Next>>](#) button.
7. A copy of the clause(s) you selected will quickly be assembled into a new document.
8. If your now assembled document contains <<*Optional*>> or <<*Options*>> or <<*Repeats*>> blocks, they are automatically processed. Once processed, you will have a final draft of the document. If the document contains variables (placeholders for personal data), you would then activate the Instant Database system to either recall an existing data record for the client or customer, or create one. The Instant Database routine is activated by pressing Alt-D (for 'D'atabase).

Many users start the document assembly process from what are commonly called 'templates.' These template are essentially complete, if not overbuilt, documents. While the above steps will let you access a single document in order to create it, it is not the most efficient way to do so. Using a DropDown List would be much more efficient. See [Technique 2](#)⁴⁵ in the next section.

See also:

In General:

[Libraries & Books Screen](#)⁴⁶

[Clause Selection Screen](#)⁵²

[Preview](#)⁶²

Document Assembly Alternatives:

[DropDown Lists](#)³⁵⁶

['Margin' Lists](#)⁴²⁶

[<Alt-G>](#)⁹⁴

[Clause Sets](#)⁴⁰⁶

3.2 The Process, Technique 2

Template Assembly:

Rather than using the 'paragraph assembly' method described in Technique 1, many users prefer to start the process from what are commonly called 'templates.' Templates are essentially complete, if not overbuilt, documents. They don't offer quite as much flexibility in terms of the possible number of document permutations, but do offer greater control over the content the end user can select. Further, they are typically easier to create, use and conceptually understand for those just beginning to use the program.

There is no special preparation that is needed in order to 'create' a new document based on a template. The only required action is to assign the folder containing the templates to either

- a 'book.' This is preferable if you tend to select document in collections. (Here we are talking about 'packages' of documents as opposed to a single document. When assigned to a book, all templates in the folder can be displayed in the Clause Selection Screen, where one or more templates can be selected and assembled.
- a DropDown List. This is preferable when the office practice is to create one document at a time from a template. Click at the link for the steps to create a [DropDown List](#)³⁵⁶. Once accomplished, document assembly is a simple matter of 'point and click:' Select the DropDown list containing the desired document and click the desired document.

A copy of the document you selected will quickly be displayed onto the editing screen. (Note that we said a 'copy' is displayed. It has all of the attributes of the original, but since it is a copy, there is no possibility that you will accidentally overwrite the original version.

If your now assembled document contains <<*Optional*>> or <<*Options*>> or <<*Repeats*>> blocks, they are automatically processed. These blocks allow you to select among choices that the document author has placed in the source text. Once selected, you will have a final draft of the document. If the document contains variables (placeholders for personal data), you would then activate the Instant Database system to either recall an existing data record for the client or customer, or create one. The Instant Database routine is activated by pressing Alt-D (for 'D'atabase).

See also:

[DropDown Lists](#)³⁵⁶

[<<Options/Optional*>> Text](#)¹⁴¹

[<<Repeat>> Blocks](#)³⁸²

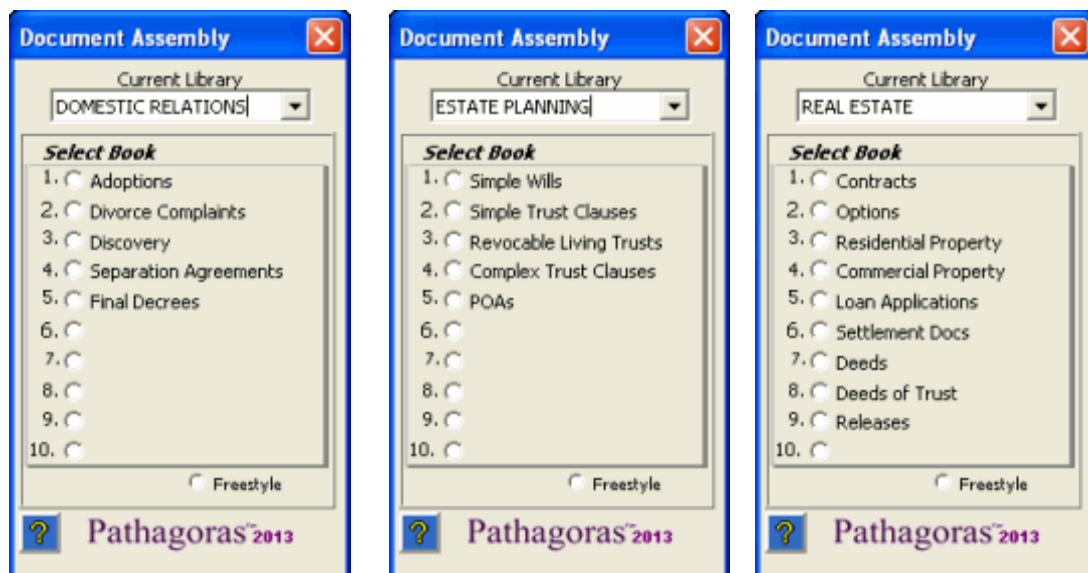
3.3 Libraries & Books: Concepts and Definitions

Whether you adopt the 'clause assembly' or 'template' approach to building documents, you should set up a Library and shelf several books within that library early on in your Pathagoras experience. This will help you to quickly access the documents and clauses you need to create your documents and confirm how valuable a document assembly program can be to your practice.

Pathagoras follow a 'libraries and books' metaphor for storing and accessing documents and/or clauses used to create final documents. This metaphor is discussed and more fully illustrated at this link. Essentially the user (or system administrator) would create a 'Library' and assign up to up to 10 books to that library. In a classic setup, the libraries bear names that reflect a general subject area (such as 'Estate Planning' and 'Domestic Relations'. The various books that the user would 'shelve' in the library would reflect more 'specific topics.'

Your entry point for the 'classic' method of assembling a document starts with the **Document Assembly (Libraries & Books)** screen. (Click here for instructions on how to display the screen.)

Conceptually, here is what several Libraries, each with several books, might look like:



Once you have a document assembly objective, click the Document Assembly icon in the Word toolbar area. Select a library by dropping down the 'Current Library' dropdown. Then, with the desired library selected, choose the type of document you wish to create by clicking the appropriate book. When the clauses within the book display only the Clause Selection Screen (discussed below), select one, several, or all of the clauses you want your final document to contain. Click the appropriate 'assemble' button and you are done. It is really that easy.

Concepts to keep in mind. A book is merely a pointer to the folder that you assign to a particular shelf. A library is simply a collection of those pointers.

For example, the book 'Adoptions' in the Domestic Relations library example above points to the folder that contains your adoption documents and clauses. You can see the name of the folder to which the book points by hovering your mouse over the book name. Example:



With the mouse hovering over a particular book, the folder to which that book points shows in the 'tip text' block

Neither the Adoptions book or any other book is 'owned' by Pathagoras. Pathagoras simply uses what you have in place. (As you can see, the 'Adoptions' folder is not a 'Pathagoras' folder. It pre-existed Pathagoras. Pathagoras doesn't require you to change a thing. Of course, it always is advisable to organize your documents in a logical structure within Word and Windows, such as the above illustrates, but there is no requirement that you do so. Even dis-organized filing systems can be made much more accessible via the libraries and books system offered by Pathagoras.)

Read more about the nuts and bolts of using the Libraries & Books screenage in the following pages.

3.4 Libraries & Books Screen_2

Your entry point for the 'classic' method of assembling a document starts with the **Document Assembly (Libraries & Books)** screen. (Click here for instructions on how to display the screen.)

The following screen shots illustrate the various elements of that screen. (The illustrations reflect the DocAssemDemo library that ships with Pathagoras). The details of each section are explained further below.

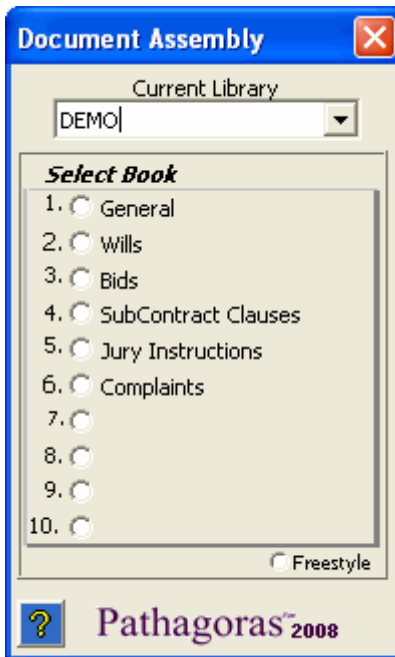


Figure 1. The initial screen. Select the library, and then select the book which contains the clauses you wish to assemble.

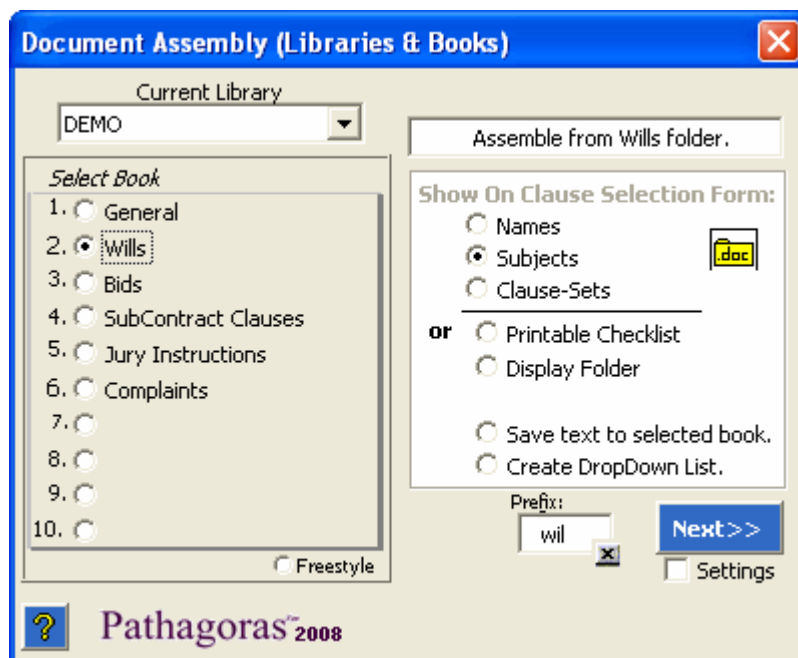


Figure 2. Options now appear.

After you choose a 'book' from the initial screen, the screen enlarges to provide more options*.

 ***Note:** The expanded screen does not appear when 'End User' settings have been selected. The user is taken directly to the Clause Selection screen. See [End User Settings](#)



- **Action Confirmation:** At the top of the new section, Pathagoras confirms the selection that you made. In the example, we chose the Will folder. The top line indicates that the 'Wills folder' is selected for assembly.

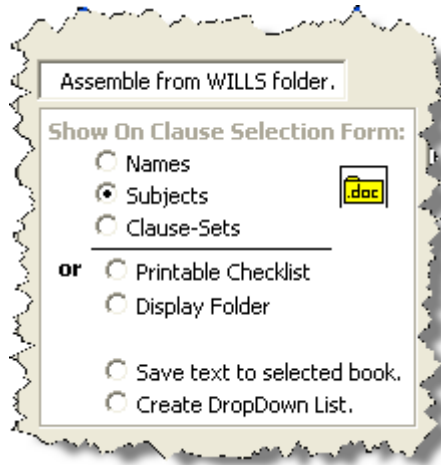


Figure 3. The 'expanded' section of the Libraries & Books screen

- **Show on Clause Selection Form:**

- **'Names' or 'Subjects':** You can display the terms on the clause selection screen by the 'Name' or by the 'Subject' assigned to each term. The document Name is typically a short few words. The 'Subject' is much more descriptive. (If you study the 'Will' and the 'SubContract Clauses' books, you will see that the clauses in those books use the more cryptic prefix/suffix pattern for names. More on this below and in other writings.)
- **'Clause-sets'** are collections of selected clauses that represent complete documents, but which can be called into your document by a single name. Clause-sets are great for standard contracts, simple Wills, etc. Instead of choosing the same X-number of clauses for a standard document, create a clause-set and select just one item. See [Clause-Sets](#)

NOTE: To the right of the 'Name' and 'Subject' radio buttons in the center panel, you will see a symbol like one of the following:



These symbols visually depict the nature of the clauses in the selected book. Click on each of the various books, and you will see the symbols change as you work your way down. A folder with the letters '.doc' signifies that the book contains standard Word documents. A folder with the letters '.dot' indicates that the terms are stored in the folder as Word templates. A folder with the letters '.txt' indicates that the terms are stored as unformatted ASCII text files. (Pathagoras can assemble those too!) A Glossary is denoted by the grey square box with the 'G' in the lower left side. 'G' = glossary.

- **Administrative tools** reside 'below the line.' These tools allow you to perform certain operations on a selected book


(1) **Printable Checklist:** You can create a checklist of all of the terms in the selected book. Use the list just for a handy reference, or print it out as a check-list and 'check off' the desired terms for the new document.)

(2) **Display Folder:** You can open the folder or the glossary if you need to quickly access either for editing or other management purposes.

If the book is a folder of documents, you will be taken to the Windows folder that contains the terms. Everything in here should look familiar. These are simple Word documents. Nothing more, nothing less.


If the book is a 'glossary,' you will be inside the actual glossary document itself. It is a rather simple document. It contains a 'red line' that reflects the term's 'name,' a blue line that reflects its 'subject,' and the actual text of the term immediately following. While you may not be able to see them, each term is enclosed within 'bookmarks' that Word (and Pathagoras) uses to identify the begin and end points of the term. Despite its structure, a glossary is still a just a Word document.


(3) **Save text to selected book:** If you wish to save the currently displayed document (or a highlighted portion of that document) into the selected book, select this option. Press Next. Pathagoras will call up the *Term Works!* screen. There, give the new term/document/clause a name and a subject. Pathagoras will place it into the folder or glossary represented by the book. Little could be simpler. No navigation required.


(4) **Create DropDown List:** You can create a drop-down list of the terms in the selected book. These are very handy, very powerful, always-on, always-active lists that reside at the top of your editing screen which allow you to select any term or form by pointing and clicking. Read more about these very powerful [DropDown lists by clicking here.](#) 


Settings:

In the lower right hand side of the screen is a button labeled "Settings." Additional setup and administrative tools reside there to augment and personalize the program to your needs. Here are some of the functions:

Add new book into the library, see [Adding a New Book](#) 

Remove book from a library, see [Removing a Book](#) 

Add new library to your system, see [Creating a New Library](#) 

Delete existing libraries from your system, see [Deleting a Library.](#) 

ShortCuts:

Set a book as the "default" book: *Shift-Click* on the book

Repoint a book without going through the Settings screen: *Ctrl-Click* on the book.

See Also:

[Default Settings](#) 

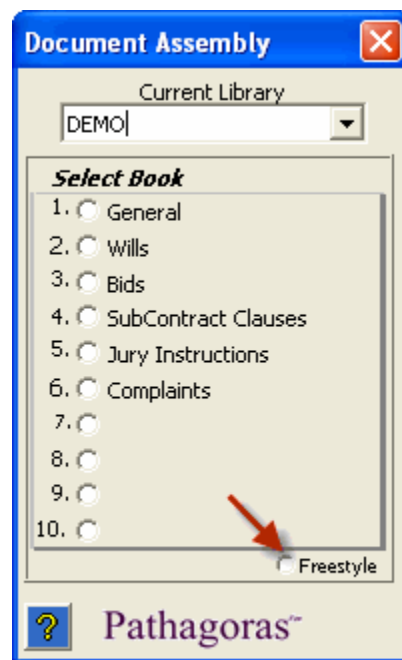
3.5 'FreeStyle' Display

The previous sections contemplate that you have 'shelved books' into your libraries, and that all folders that you need to effectively use your document assembly system are mapped to an appropriate book.

Shelving books into one or more libraries is always the preferred method in setting up your system. You can have an unlimited number of books by simply creating more libraries. No 'space' is consumed by having more books.

Nevertheless, there may be a need to access a particular folder that has not been mapped. Pathagoras provides a 'freestyle' way to display the contents of any folder on your system (local or network) onto a Clause Selection Screen. The process is actually quite similar to the 'normal' clause selection process:

1. Click the Document Assembly icon from the Pathagoras toolbar. The Libraries & Books screen will appear (image below).




2. If the folder has never been mapped to a book, click the Freestyle option button. Click the Next button from the expanded screen. Following the prompts, navigate to the desired folder and select on document from that folder. **IT DOES NOT MATTER WHICH DOCUMENT YOU SELECT.** The exercise is to get inside the folder (so you know you are in the correct one. Selecting a document merely locks in your folder selection and makes the OK button 'hot' so that you can proceed.)

Once you click the OK button following your navigation to the desired folder, Pathagoras will display onto a Clause Selection Screen (left panel) all documents in that folder. Select one, several or all of them and move them to the right panel. Click Next and the selected documents are assembled.

(Of course if the folder you want is in the 1 thru 10 list, select it. If the folder is mapped to

a book in another library, drop down the library list and select the appropriate one and proceed.)

 **Note:** The Freestyle option does not appear when 'End User' settings have been selected. See [End User Settings](#) ⁵⁷⁵.

3.6 Clause Selection Screen

The Clause Selection Screen is the place where you choose which of the book's clauses, forms or documents will be assembled into the final document.

The Clause Selection Screen appears after the operator selects a particular book from the Libraries & Books screen and presses **Next>>**. A two-panel form similar to the ones displayed below will appear.

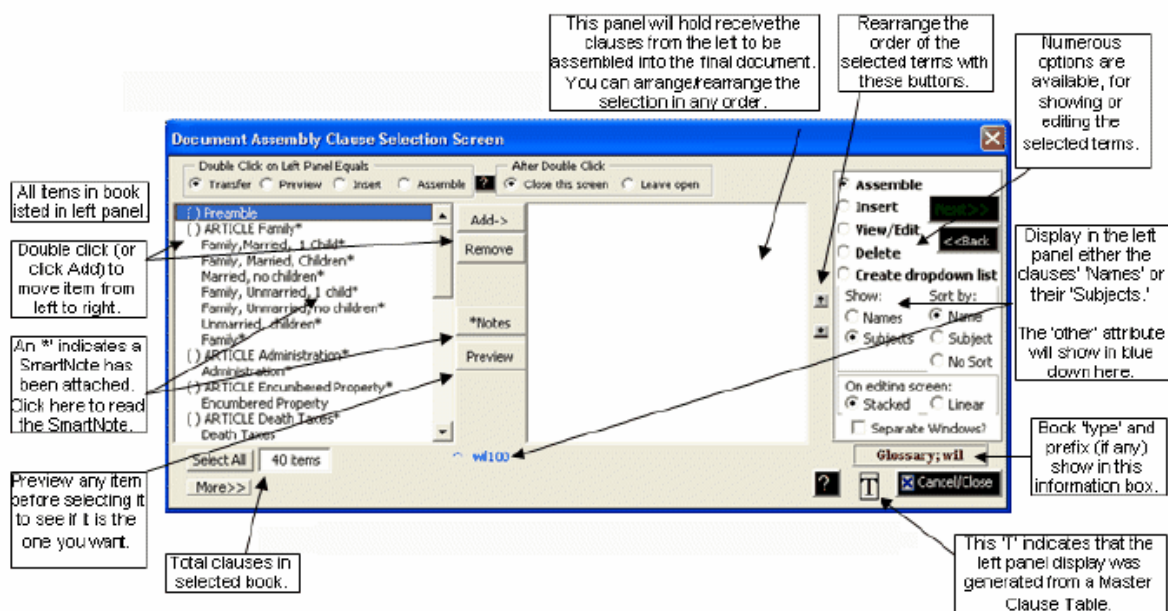


Figure 1. The 'normal' Clause Selection Screen.

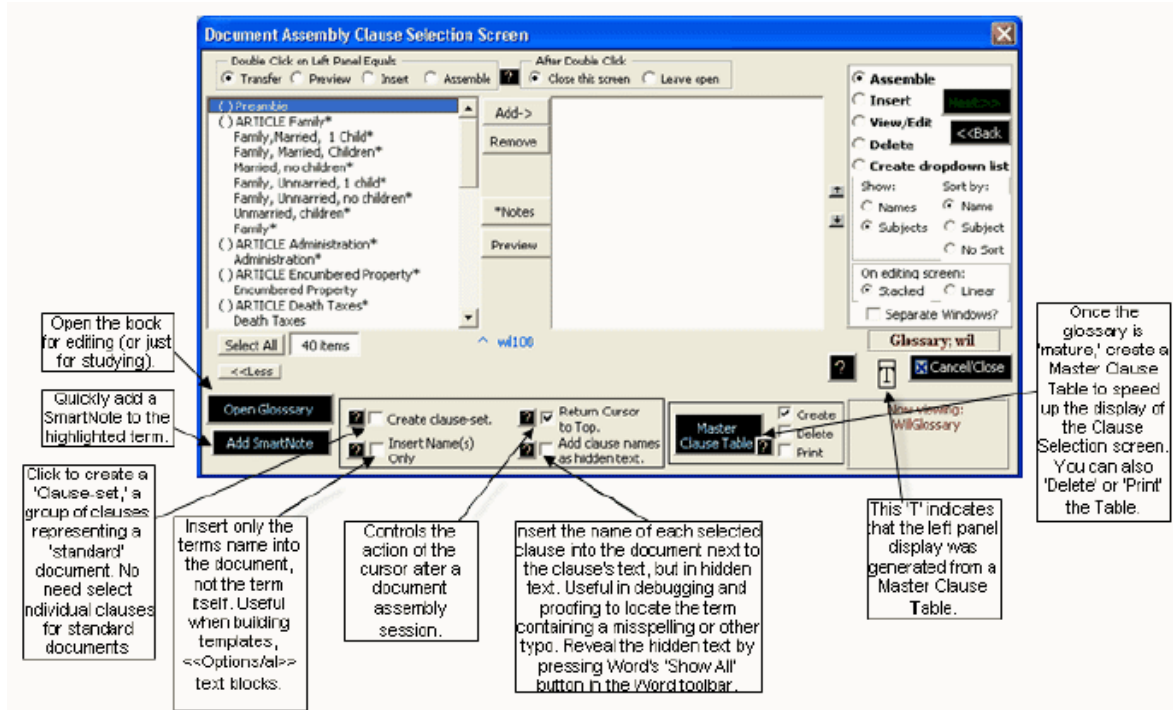
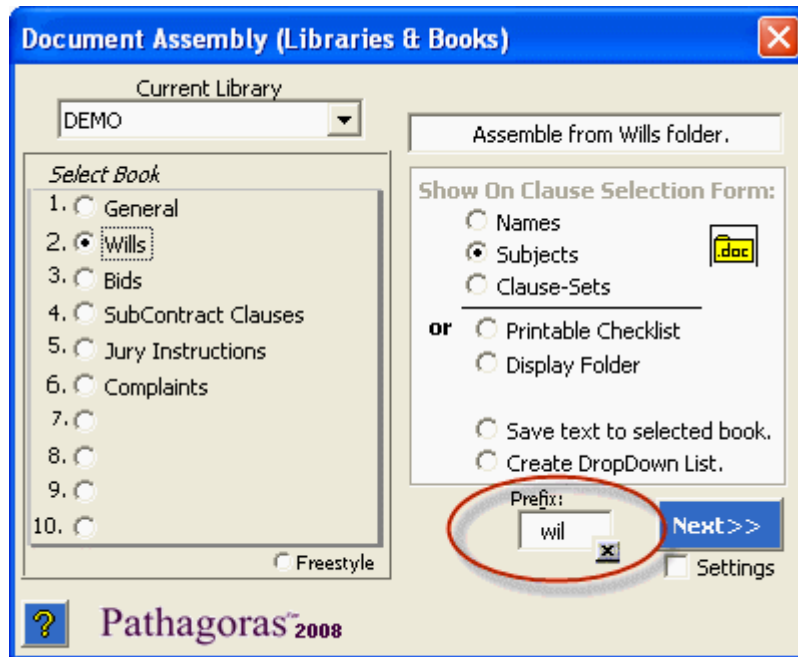


Figure 2. The Clause Selection Screen with the 'More' features showing.

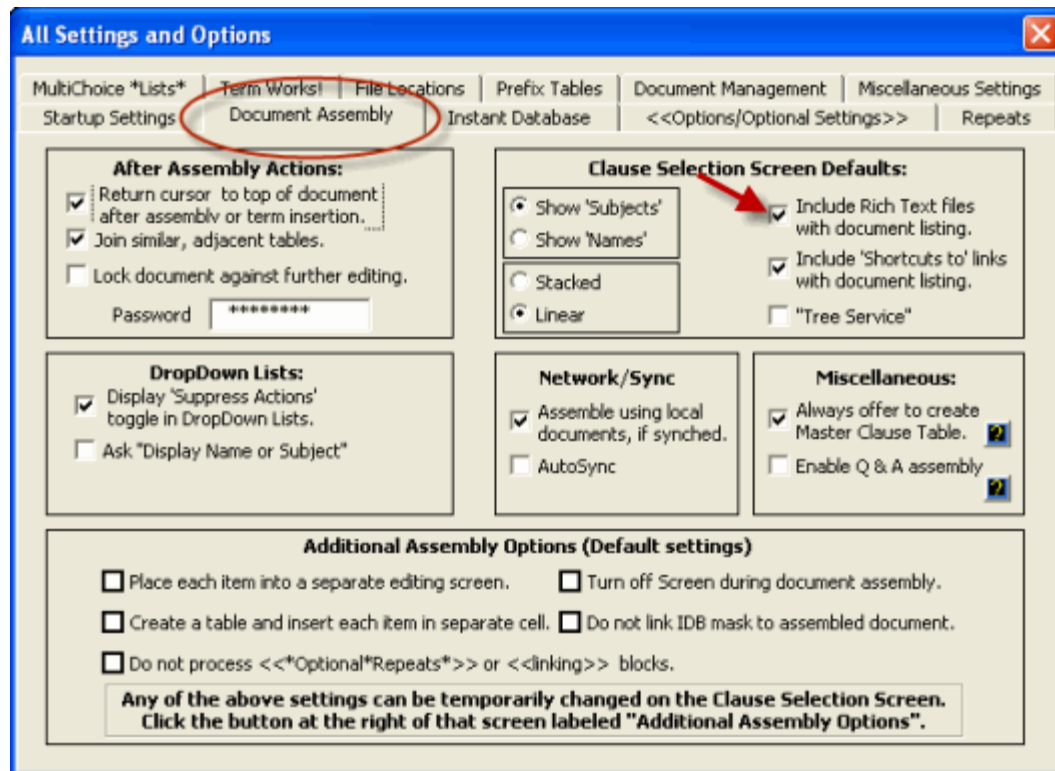
3.6.1 Files Included in Display

The Clause Selection Screen will typically display all of the documents in the folder or the terms in the glossary that has been designated at the 'book.' You can cause the listing to be augmented or restricted in several fashions:

Prefix: on the initial Document Assembly screen (we call it the Libraries & Books screen), look for the 'Prefix' list to the left of the Next button on the expanded screen. You can keep the prefix, and the subsequent listing will show only those clauses which begin with the prefix, or you may select another prefix (do so by typing in the prefix or selecting one from the dropdown that will appear if multiple prefixes have been assigned), or you may delete the prefix (click the 'x') and show all terms in the book.




RichText files: The default scope of listing of the content of a folder is "*.doc" and "*.docx" files. You can include RichText (*.rtf) files for automatic inclusion in the Clause Selection Screen display. To do so,



ShortCuts: The default scope of listing of the content of a folder is "*.doc" and "*.docx" files, but you can include shortcuts to documents located in other folders. 'Shortcuts' can link to anything, including documents. Pathagoras can decode those links and, if it points to a

document, can display in Pathagoras Clause Selection Screen the target of the links as if they were actual documents within the folder.

i ShortCuts are those devices which you frequently encounter on your desktop or which you see in folder listings that will (1) quickly transport you to another location, (2) call up a document that actually resides in another folder or (3) call up the program to which it points.

Shortcuts are indicated by a little arrow on the face of the icon that otherwise represents the target. For Pathagoras purposes, we are primarily talking about shortcuts to Word documents. The shortcut icon is the same as the Word document icon, but with a curved arrow in the lower left quadrant, thusly: 

Using Shortcuts: Let's say you have several folders each containing narrowly focused documents on a specific topic. You place the documents related to topic A into Book A, those related to topic B into Book B, etc.

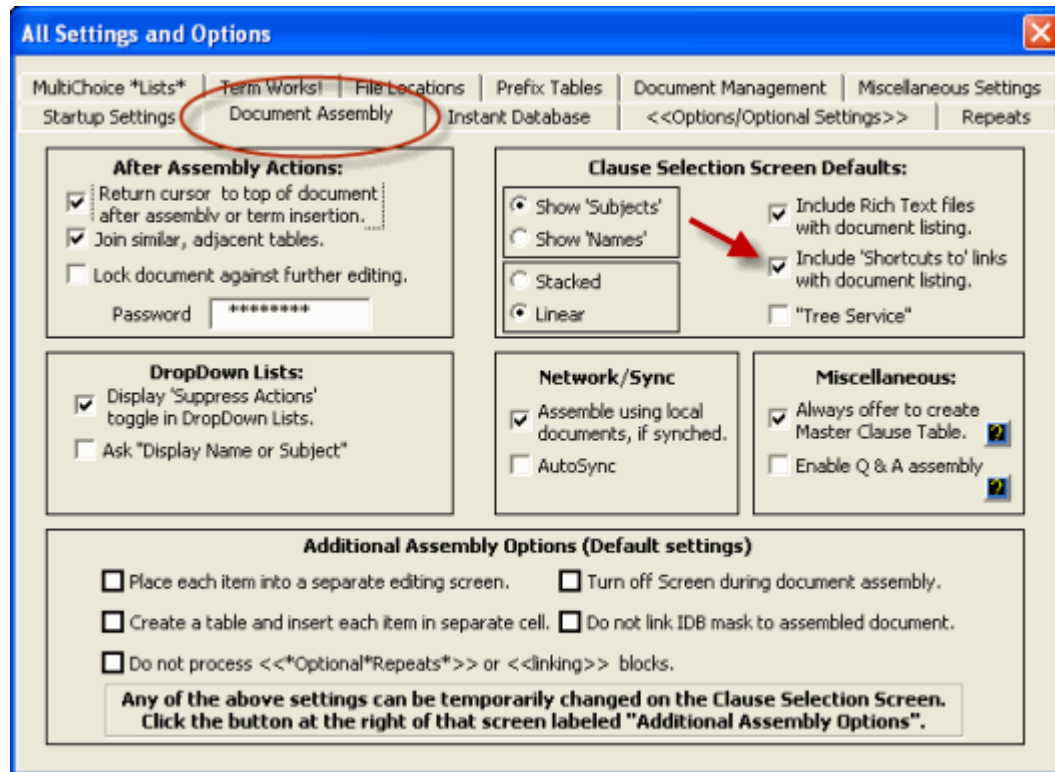
Let's further say that you have a folder of 'generic' documents. Because they contain 'universal' type text that you may use in a variety of situations, you want those documents to display on the Clause Selection Screen along with the 'A' documents whenever you display Book A, with the 'B' documents when you display Book B, etc.

To accomplish this, navigate to the folder containing the generic documents. (This is simple Word/Windows stuff.) Highlight all of the documents/clauses for which you wish to create shortcuts. Right click and select "Create Shortcut". Windows creates the shortcuts and places them at the bottom of the display screen.

Next, copy and paste the shortcuts into the folders represented by Book A, Book B, etc. (When you are all done with the exercise, you will probably also want to delete the shortcuts from the original location.)

Now, whenever you display Book A or Book B, etc., through Pathagoras' document assembly routine, the 'native' documents AND the shortcuts will display. (Any shortcuts in the target folder will display *after* the native documents.)

Document shortcuts will display by default, but if you find them distracting, you can elect not to display them by unchecking the box called "Display 'Shortcuts to' links". You can locate this checkbox under "Utilities & Settings | All Settings | Document Assembly (tab) | Clause Selection Screen Defaults (section)".



3.6.2 Buttons and Functions

What follows is a summary of the buttons you can press, and the choices you can make, from the Clause Selection Screen.

3.6.2.1 Add and Remove

Two large panels command the center of the Clause Selection Screen.

- The left panel contains a listing of the clauses in the source folder or glossary. Depending upon whether 'Name' or 'Subject' was selected in the Libraries & Books screen, the list will show the appropriate attribute of the document. The other attribute will show in blue at the bottom of the screen, between the two columns. (If a filter was chosen in the Libraries & Books screen, only the items meeting the filter settings are shown.)
- The initially empty right pane, will receive the clauses you select from the left panel. The clauses that are present in the right panel when you press Next>> are the ones that will be assembled.

Move desired clauses from the left panel to the right by:

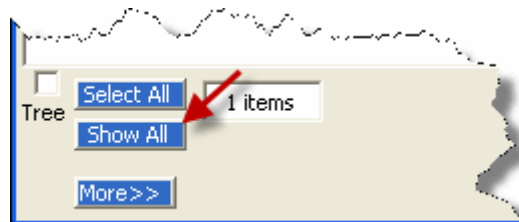
- selecting an item and pressing the Add-> button. (Multiple selections can be made using the Shift- (adjacent) and Ctrl- (non-adjacent items) keys; or
- double clicking on an entry.

Remove items from the right panel by clicking the <Remove> button.



3.6.2.2 Show All

The "Show All" button (toward the lower left side of the screen) allows you to repaint the left "Documents Available" panel to display every file in the parent folder. (The default display is the show just documents or glossary terms that meet the filtering criteria set in the "Libraries & Books" screen. This not only will display all documents that 'exceed' the filter, but it will also display non-document files such as Excel files and images that you have saved in the designated book. This in turn enables Excel and image assembly as well as paragraph and document assembly.



3.6.2.3 Select All

If you wish to select all (or almost all) of the available clauses, click the Select All button (located near the bottom left of the screen).



3.6.2.4 'Rearrange' arrows

If you wish to rearrange the order in which the selected items will be assembled, highlight the clause you want to move and use the Up and Down arrows to the right of the right-hand

panel to establish the proper order.

3.6.2.5 Forms 'Packages'

'Packages' are collections of related documents that are typically prepared as part of a single transaction.

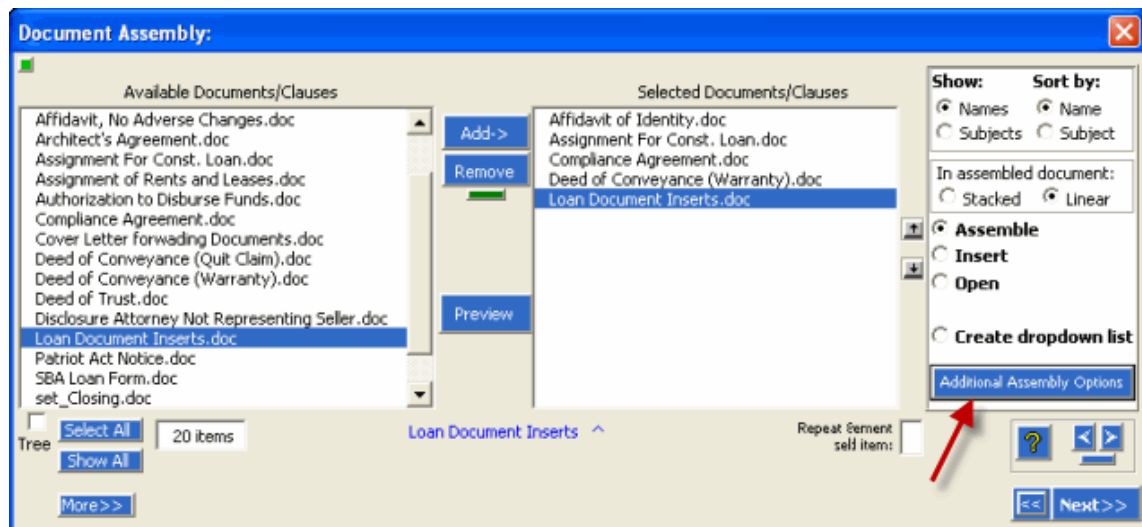
Packages and Clause Sets are sibling concepts. They both involve multiple elements to form the whole. Typically, clause sets are individual clauses that, when combined, create a single document.

Packages, on the other hand, are individual documents that, when combined, provide the documentation needed to complete a particular transaction.

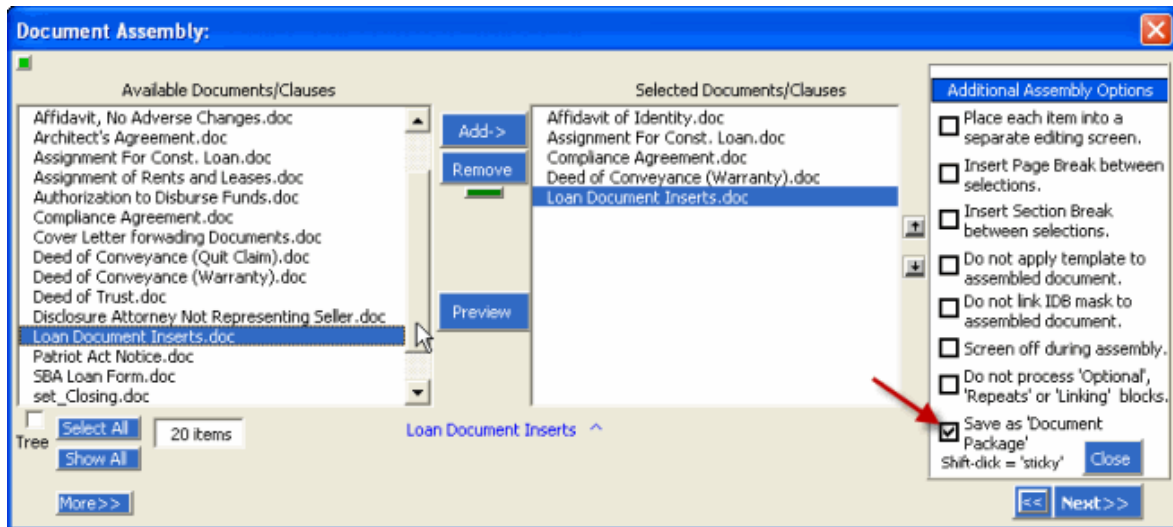
Packages can be created in several different ways. One 'non-Pathagorean' method is simply to navigate to the folder where the essential documents now reside (the 'source' folder) and copy the documents desired for the package. Then, navigate to a client folder (or create a new folder for a new or existing client) and paste the selected documents into the target folder.

But Pathagoras can perform the manual steps described above for you automatically. The only requirement is that the source folder be assigned to a book in one of your libraries. Assuming that you have done that, follow these steps:

1. Click on the Document Assembly icon.
2. Select the library and then book that contains the desired documents. Click 'Next' to display the clause selection screen.
3. Select (by moving from the left panel to the right) the documents that will make up the package.
4. Click the 'Additional Assembly Options'



5. . . . and then check the 'Create Document Package' box.



6. Click the Next button. You will be prompted to navigate to client folder (or to create a new folder for a new or existing client). When you have done so, Pathagoras will (1) copy identical copies of the selected documents into the designated folder and (2) prompt you for further processing instructions. Follow them as appropriate.

Advantages of 'packages':

- Each document maintains all of the formatting and layout elements of the original (because it is an exact copy of the original). Therefore, you avoid formatting issues that are otherwise caused when documents of disparate formatting styles and layouts (e.g., fonts styles, portrait vs. landscape orientation, etc.) are combined as a single document.
- You do not have to provide names to the final documents. They always carry the name of the original documents. But since they are saved in a new folder, you never have to worry about accidentally messing up the originals. (Some customers like to name the new documents with the client's name in addition to the original name. Pathagoras provides that option.)

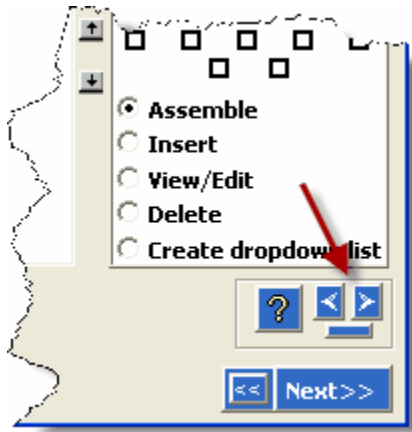
Note: If you use the same package of clauses with some frequency, you should consider creating a 'Clause Set' and then running the Clause Set as a Document Package. This is a really 'cool' combination of the two concepts. [See this section of the Manual](#) ⁴¹⁴ on how to create and then run a Clause Set as a Document Package.



If you create 'Packages' often, you can default the Document Packages button to 'checked' by a shift-click on the checkbox.

3.6.2.6 Widen Screen

If the Clause Selection Screen does not show the full name of the possible clauses, you can widen the display panels by clicking the 'Wider' button in the lower right quadrant of the screen.



Click the <Wider> button to enlarge the display panels of the Clause Selection Screen.

3.6.2.7 Assemble vs. Insert

The '**Assemble**' and '**Insert**' radio buttons at the far right side of the screen control the functions before the first selected item is process.

- If '**Assemble**' is selected, Pathagoras creates a new document and then pours the selected clauses into the new document.
- '**Insert**' pours the selected clauses into the existing document at the current location of the cursor.

3.6.2.8 View/Edit, etc.

The remaining options work on individual clauses highlighted in the right panel.

View/Edit: Opens the source clause or document. Use this for editing the original material.

Delete: Deletes the source clause. Use with caution. (A confirmatory warning is presented to prevent accidental use of this feature.)

Create DropDown List: Allows you to create a highly refined list of items for one of your Pathagoras DropDown Lists. Only the items in the right panel will be included. See also: [DropDown Lists](#) ¹²⁹.

3.6.2.9 Names vs. Subjects

Pathagoras maintains two attributes regarding each document or clause.

- '**Name**' is the obvious one. All documents have names. It is the actual document name to which this attribute refers.
- '**Subject**' Sometimes a document's name will have little meaning to the operator. Pathagoras takes advantage of the property field called '**Subject**' (meta data within every Word document) to maintain a 'better' description of the document or clause.

Pathagoras can readily display both the name and the subject assigned to a document. The radio buttons found on both the Libraries & Books and the Clause Selection Screen allow you to quickly switch between the two. (If no subject is assigned to a document, Pathagoras will 'repeat' the document's name as the Subject when that option is pressed.)

The Clause Selection Screen actually displays both the Name and the Subject simultaneously. If **Name** is the primary display, the Subject is displayed in blue text at the bottom of the screen just beneath and between the two panels. If **Subject** is the primary display, the clause's name is displayed in the blue text.

See Also:

[Adding/Editing Subjects](#)  320

[Names & Subjects](#)  320

[Editing Names & Subjects \(en masse\)](#)  428

[Libraries & Books Screen](#)  46

[Clause Selection Screen](#)  52

[Preview](#)  62

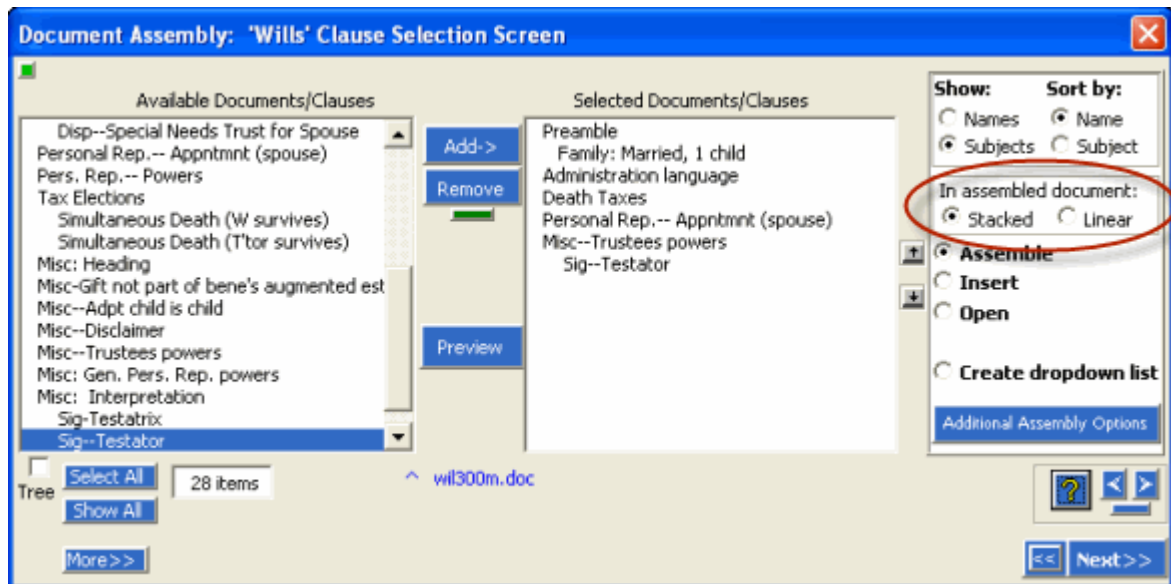
3.6.2.10 Linear vs. Stacked

When Pathagoras assembles a document using 'paragraph assembly,' it will (by default) insure that a paragraph marker is inserted after each requested clause. This model of assembly is called 'stacked'. Each new clause is inserted below the previous one in a 'stacked' fashion.

Perhaps the assembly is not of paragraphs, but rather the clauses are sentences or single words and the author intends for no paragraph makers to separate the terms:

STACKED	LINEAR
The available colors are: Red Blue Green	The available colors are: Red Blue Green

You can select/change the insertion method by selecting the appropriate option at the right side of the Clause Selection Screen:



3.6.2.11 Preview

You can preview the text of any clause before deciding to select it for assembly. Click the <Preview> button found between the two columns of the Clause Selection Screen. Pathagoras will display in a separate screen the first 1000 characters in the previewed element.



Press the Preview button on Clause Selection Screen to view the first 1000 character of selected item.

3.6.2.12 Expand (Clause Set)

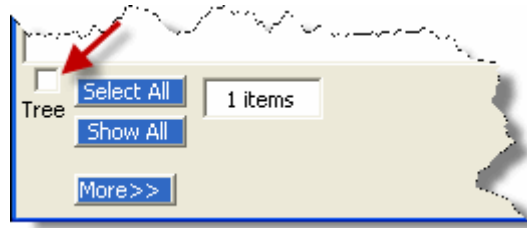
If you wish to view the individual components of a [Clause Set](#)⁴⁰⁶, click the <Expand> button that will appear between the two panels of the Clause Selection Screen when a clause set is highlighted.

➔ Once 'Expand' has been clicked, the resulting list can be used and manipulated in the same fashion as if you had manually inserted the clauses individually. You can rearrange the clauses, remove any of them, and add to them from items listed in the left panel.

3.6.2.13 Tree Service

The default display for a book is all documents in the folder assigned to the book. However, you can display (and therefore choose from) clauses in the parent folder and all clauses in folders beneath the parent folder. Just click the Tree checkbox. The left panel screen will be

repainted, adding to the top of the list any sub-folders beneath the parent folder. A sub-folder will be displayed between {curley braces}. Double click on the sub-folder name to display its contents.

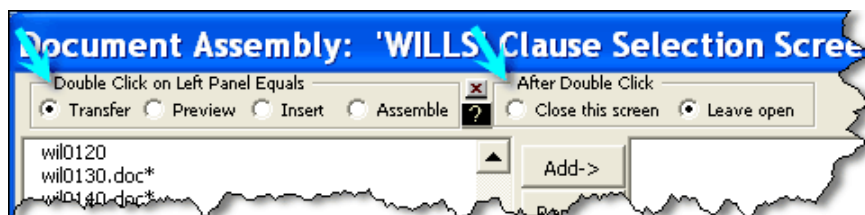


3.6.2.14 Double Click Tools

Preliminary Note: Unless the feature is already activated, you should see a small green button in the upper left side of the Clause Selection Screen. Click it if you wish to activate the features discussed below. If you choose not to activate the Double Click controls, the action of a 'double click' of the mouse on a selected entry in the left panel is to transfer it to the right panel.

Using the double click tools is *entirely optional*. It is intended for more advanced users.

If activated, the buttons along the top left edge of the screen control what will happen when you double click on an entry in the left panel. There are two separate control sets.



The Clause Selection Screen w/ Double Click Menu activated.

1. Left Side options:

- 'Transfer' (default). Transfer an entry from the left panel to the right. (Identical to pressing the Add> button. The default)
- 'Preview:' This will display the first 1000 characters in the document or glossary term.
- 'Insert': places the selected term at insertion point of active document. (You can actually build a document via the Clause Selection Screen using this selection.)
- 'Assemble' places the next selected item into new document. The selection automatically reverts to 'Insert' afterwards.

2. Right Side options:

- As the terms suggest, the 'After Double Click' options determine the status of the Clause Selection Screen after you make the double-click selection.
- You would probably choose 'Close this screen' if you are selecting a single clause, or a clause-set.

- Choose 'Leave open' if you intend to make several selections.

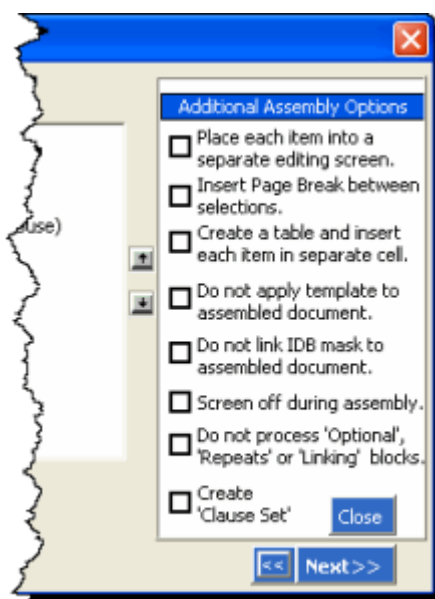
The 'x' in between the two sections dismisses the double click screen. The '?' will provide additional guidance.

3.6.2.15 Assembly Options

At the bottom of the action panel is a button labelled "Additional Assembly Options." Clicking it will reveal a series of choices which can greatly enhance your document assembly results. Some boxes control layout issues. Other help as you are refining and testing your system. They allow you different views of your assembled text based on different parameters.



Typically, all boxes will be unchecked, but when you need a document assembled using different rules and views, this is where it can be accomplished.



NOTE: When you hover over any of the checkboxes, a pop-up tip will appear reminding you of its function.

From top to bottom:

1. Insert each selection into a separate document. If you select 5 clauses, you will end up with 5 separate documents as each selection will become a separate document. This is an alternative to assembling all documents in a single screen, replacing variables and then disassembling the document into component pieces. It is especially useful when the individual documents use different formatting or style definitions, different margin settings, page orientations or other settings that make assembling a 'huge' document impractical.
2. Create a single document, but insert page breaks between each selected clause. That way, each clause will start on its own separate page. (This is different from adding Pathagoras' "&New Doc&" marker. The goal of inserting "&New Doc&" is so that you can ultimately break the document into multiple documents on separate editing screens. In such case, each can be saved as a separate document. But see #1 above for an alternative.)
3. Create a table. Insert selections into individual cells. Pathagoras will pause to ask how many columns you desire. It calculates the number of rows needed by simple division.
4. Ignore assigned template. If you want to view an absolutely 'pristine' document built using the 'normal' template, check this box. Assignment of templates to documents created from a selected book is made in the Settings screen. (Look for the Templates tab.)
5. Ignore Instant Database Mask assignments. When a mask is automatically assigned to your assembled document, Pathagoras will not scan the document for variables during an Instant Database session. Rather it will present the mask. If you have document variables that you need to 'discover' without the interference of a pre-linked mask, check this box. (Assignments of Masks to books is made in the Document Assembly Settings screen. Look for the Auto-IDB tab.)
6. Turn off screen display during assembly. Pathagoras tries to show as little screen activity as possible as a document is being constructed. But screen flashing and repaints will inevitably occur during the assembly process. With short documents, the visual effect is typically not troublesome nor slow. For extremely long documents, the opposite might be true. If you click box #5, Pathagoras will quite literally 'hide' Word, assemble the document and when done, return to Word with a complete document. Whether to assemble in this way is a personal choice. (It does take some getting used to. Seeing Word 'disappear' can be disconcerting at first.
7. Do not process <<*Options/Optional*>> text blocks after assembly. All such blocks will remain in the draft. Good for testing, or for when multiple calls on a document assembly routine must be performed before processing should occur. (To process the document when it is not automatically triggered, press <Alt-P>.
8. Create a [Clause Set](#)⁴⁰⁶ of the selected clauses. (Same result as if you clicked "More | Create Clause Set."

3.6.2.16 Add 'Hidden' Clause Names to text

As you review assembled documents, you may discover a typographical or formatting error that you wish to correct. But you may not know which clause is the one that contains the problem text. In order to solve that 'mystery,' you can tell Pathagoras to add the names of each clause that you have selected for assembly as a piece of hidden text preceding the insertion of the clause itself. Here is how:

1. Select the desired book of clauses from the Libraries & Books screen and press **Next>>** to display the Clause Selection screen;
2. Select the clauses that you want to assemble and move them to the right pane. Before clicking the Next>> button to begin the actual document assembly, click the More>> button in the lower left side of the screen. Several additional controls display, including "Add clause name as hidden text." (See Figure 1.) Check it.

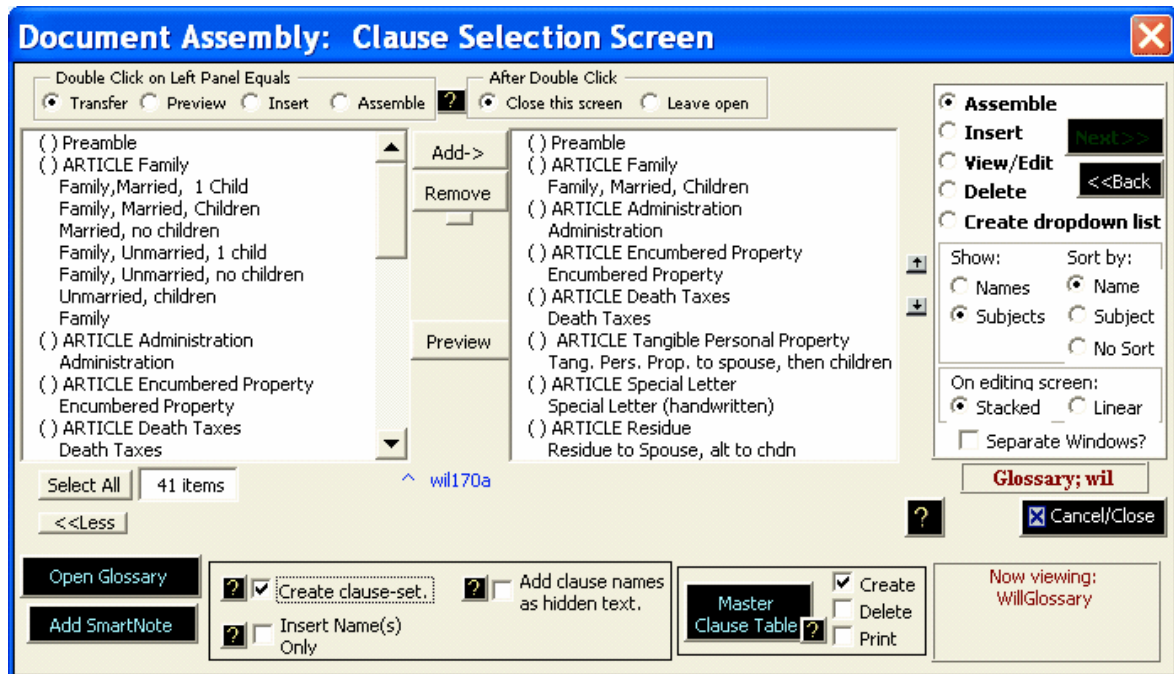


Figure 1. The Clause Selection Screen ("More" view)

4. Now press the **Next>>** button in the upper right corner of the screen. The names of each selected clause is inserted into the new document as hidden text immediately before that clause's text. To 'reveal' the clause name, press the 'Show All' button (the pilcrow character) in the menu area of the screen.

3.6.3 Master Clause Table

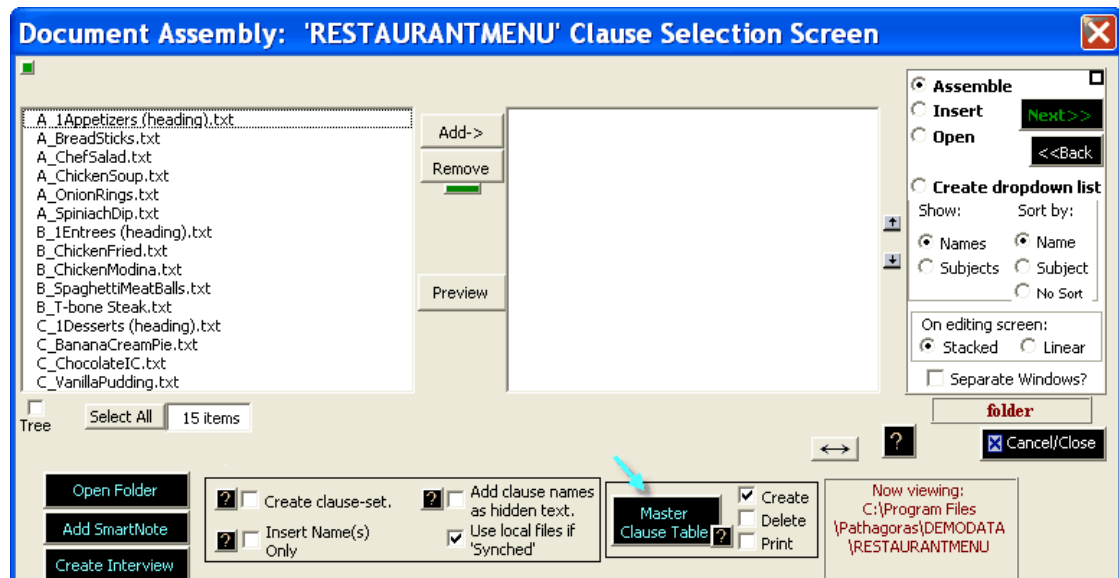
Pathagoras is a highly dynamic program. Every time you call for a display of a book in the Clause Selection Screen, Pathagoras will scan the contents of the book (i.e., the entire folder or glossary) anew. It will read both the name and subject of each clause. That way, Pathagoras can insure that every clause in the book (even ones that you may have just added) is always presented in the left panel.

For large glossaries or folders with many documents, or if you have a relatively slow computer, the process of reading the names and subjects of each item in the book can take several seconds. If your glossary or folder of terms is large but stable (i.e., you are no longer adding terms to it), you should consider creating a 'Master Clause Table' for that particular book.

When Pathagoras detects the existence of a Master Clause Table, it will use that table to populate the left panel in the Clause Selection Screen, significantly speeding up the display of the Screen.

To create a Master Clause Table:

1. Display the contents of the book to the Clause Selection Screen in the usual fashion.
2. Click the More>> button. This will reveal an additional set of options:



3. Click the Master Clause Table button (the checkbox reading 'Create' should be checked by default). Pathagoras will save a copy of the names and subjects of each item in the book into a separate file. It is this file that will be read to prepare the left panel list in the Clause Selection Screen.

You can delete the Master Clause Table, or recreate it, at any time.

Pathagoras signals that it is using a Master Clause Table to populate the left panel by displaying a boxed capital **T** in the lower right side of the Clause Selection Screen. If you think that there are items missing from the left panel, and if you see the boxed capital **T**, it is time to regenerate the table.

Note: If you add a new clause to a book after having created a Master Clause Table, be sure to update the table! Pathagoras is programmed with logic to check the count of items in the MCT against the actual number of items in the book, but it is best not to depend upon that check.

3.6.4 Document Assembly Issues

Spacing between paragraphs:

If you have a series of paragraphs that are coming in properly, but without any space between the paragraphs, try one of the following:

1. Choose the 'stacked' option. This will add an extra line between each paragraph as it is inserted.
2. Change the paragraph settings to add an extra bit of space between each paragraph. You can apply this setting to the style of the receiving document or (and probably better) to the style of the source document.

Order of clauses:

You can easily change the order of the clauses you select using the 'rearrange' arrow

buttons to the right of the 'Selected Clauses' panel.

Editing a clause:

Pathagoras makes it easy to access your clauses for editing. While the Libraries & Books screen, click the Open Folder option that appears after you select a book. Pathagoras quickly displays the proper folder and you can select the item you wish to edit. If you are using the prefix/suffix naming style, perhaps you know the subject of the clause, but not its name. No program. Just click Next from the Libraries & Books screen to display the Clause Selection Screen. Making sure that 'Subjects' is selected as the display, select the clause you want to edit and add it to the right panel. Click the 'View/Edit' radio option at the right and then click Next.

Inconsistent Margins and Styles:

When the margins, pages size, orientation or styles in the various documents that you are trying to assemble are not consistent with each other, you have two possible solutions:

1. Use Word 'sections' and at the top of the actual document where margins change, add a section break and re-assign the margins. Then when the document is assembled, that section break will be detected and the margins will be adopted. Make sure you do the same with the document that follows. This is not the favored solution because of the amount of setup involved.
2. Run your selected documents as individual documents. Before clicking the Next button on the Clause Selection screen, click the Additional Assembly Options button and check the first box "Place each item into separate editing screen."
3. If you have created a Clause Set containing the desired document, you can Run your Clause Set as a "Document Package". [See this section of the Manual](#)⁴¹⁴ on how to create and then run a Clause Set as a Document Package..

3.7 Document Assembly 'Books'

As noted in other sections, Pathagoras has adopted a libraries and books metaphor to describe how source clauses are arranged and stored.

A 'book' is simply a pointer to an existing location in which clauses are stored.

The concept of a Pathagoras 'book' could not be simpler. A 'book' is a pointer to folder that currently resides on your computer (or on your network). By 'pointer,' we mean only the standard Windows style 'long name' which identifies a specific folder on your system. An address like "c:\my documents\office forms\" or "g:\office forms\estate planning" and the like.

We implore you to not make this definition more complex than that. To over embellish what a book is will lead to confusion. "Book" = "pointer to folder". Nothing more.

(As you will read elsewhere, a 'book' can also be a document -- called a 'glossary' -- that contains multiple bookmarked clauses. But at least for the beginner, let's hold onto the "Book" = "folder" definition)

A "library" comprises up to 10 books. So, a library is nothing more than a collection of up to 10 pointers to folders that reside on your computer (or on your network).

The concept of a Pathagoras 'library' is likewise a very simple idea. From a technical perspective, a library is text file that Pathagoras creates to hold the pointers. This text file resides on your computer within the Pathagoras setup folders. It contains up to 10 lines of text, each line being a 'book.'

Since a library contains 'books,' and a 'book' is simply a pointer to folder, the library is nothing more than 10 lines of pointers. Line 1 of the library file contains the pointer to folder/book 1, line 2 of the library file contains the pointer to folder/book 2, etc.

When we sat down to designed the structure of Pathagoras, it was our position that less is more. Pathagoras uses what you now have. A Pathagoras book can be pointed anywhere you want, and does not require you to restructure the existing way you have organized your files.

'Shelving' a book:

The process of adding a book to a library is called 'shelving.' In reality, it is navigating to the folder to which you want the book to point. Pathagoras will automatically record of the book's address in the library when you have indicated 'this is the folder I want.'

'Types' of books:

For the novice, you will be best served if you think of 'book' equals 'folder'. But as you progress up the learning ladder, you may wish to explore the concept that a 'book' can also be what Pathagoras calls a glossary. Check out these links for for more information:

Two 'types' of books:

(1) a [folder of documents](#)⁶⁹ or

(2) a [glossary](#)⁷⁰

Each of these types of books is discussed in more detail in subsequent sections.

3.7.1 Folder of Documents

A 'folder' a standard Windows folder. Each document in such a folder represents a clause that might be used for document assembly purposes. It might be text that serves as a building block toward a larger document, or it may be a fully self-contained document. Regardless, the documents in a folder used as a Pathagoras 'book' are plain Word documents, nothing more.

We cannot overemphasize the point that the kind of documents used by Pathagoras for document assembly purposes are plain, ordinary Word files, the kind you already use, so we repeat it.

Any existing folder now existing on your computer or somewhere on your network qualifies as a book. The folder need not be one created for, or in contemplation of Pathagoras. If it is now on your system, Pathagoras can use it.

The folder that you assign to a Pathagoras 'book' can contain whole documents.

The folder can also contain various clauses needed to create (assemble) a whole document. We'll call these 'building blocks'. A clause can be anything – words, sentences, paragraphs, pages, pictures, charts, etc. Clauses can represent sequential sections of a document. Clauses can also be alternative versions of the same topic,

allowing the user to create practically infinite variety in document creation.

And, of course, the folder can contain a mixture of whole documents and building block clauses. Pathagoras is very, very flexible.

To use the files in a 'book' for document assembly purposes, you must accomplish one or both of these simple tasks:

- map it to an empty 'bookshelf.' See [Adding a New Book](#)⁷⁶
 - assign it to a Pathagoras DropDown List. See [DropDown Lists](#)³⁵⁶
-

See also [Glossary vs. Folder](#)⁷⁵

3.7.2 Glossaries

Instead of each term being saved as a separate document as in a folder of terms, they can be collected within a single document called a 'glossary.'

A glossary is a standard Word document in which a (typically) large quantity of clauses has been stored. Each clause is separated from the others by a bookmark.

The clauses in a glossary can be in effect whole documents (a glossary term can be of any length) but more typically they are the various component clauses of much larger documents.

To use a glossary for document assembly purposes, you must accomplish one or both of these simple tasks:

- map it to an empty 'bookshelf.' See [Adding a New Book](#)⁷⁶
- assign it to a Pathagoras DropDown List. See [DropDown Lists](#)³⁵⁶

A word on 'Containers'.

Everything has to be somewhere. Each clause that you use as source text in a document assembly session have to be stored in some sort of 'container.' Each container, in turn, is stored in a larger container.

When we talk about 'glossaries' vs. 'folders of clauses', we are really just talking about the *type* of container being used to store the text you want to assemble.

You already know about containers, and you know them in the same way that we are using the term 'container' in this discussion. When you draft a document on the editing screen, the characters you type onto the screen are 'contained' by the document you are creating. When you press the Save button and provide a name, the document is stored in a container called a folder. That folder in turn is probably contained by a parent folder, and that parent has a parent, all the way up to the 'c:\ drive'. The 'c:\ drive' is a subdivision of the hard drive, and the hard drive is contained within the computer box, the room, the building, etc.

A 'folder of clauses' (the most familiar of 'containers') is nothing other than a series of documents saved in a particular named container. It is nothing more complicated than that.

A 'glossary' is not too dissimilar from a folder. It too is a container. Instead of clauses being stored in individual file in a folder, the clauses in a glossary are

stored as individual clauses within a single document. The demarcation of each clause if a folder of clauses is 'a file.' the demarcation of clauses in a glossary is a pair of bookmarks. These bookmarks are named (just like a document is named) and that is how Pathagoras can find a clause in a glossary in the same way that Pathagoras can find a document in a folder.

So when you begin contemplating 'folder' vs. 'glossary', don't over-think the concepts. Just think 'container'.

See also: [Folder of Terms](#)  69

[Glossary vs. Folder](#)  75

[The 'Technical' Side](#)  71

To create a glossary, see:

[Clause Creation Tools](#)  306

[The Quickest Glossary Ever](#)  314

[Document Disassembly](#) (separate pdf pamphlet)

3.7.3 The 'Technical Side' to Glossaries

While it is not necessary to understand the 'technical' side to a glossary document in order to use one, a bit of a discussion on the care and feeding of glossaries may nevertheless be in order.

This section is intended for those who regularly use glossaries, who intend to create a new glossary from scratch or from an existing folder of files, or who are just curious as to how we got all of this stuff to work.

Glossaries are standard Word documents. That is an important initial concept to remember. Treat them as such (which means that they really are not something special, just a little different.)

Glossaries are intended to mimic the 'other' book (a folder of clauses) by housing multiple document assembly terms within its 4 corners. The individual elements of a folder are denoted by 'files' contained within that folder.

Likewise, a glossary needs to be a way to define the scope of each term. This is done using 'bookmarks.'

- A bookmark is simply a standard Word feature that allows the user to reference to a specific location or to a selection of text within a source document.
- A bookmark can be a single 'spot' in the document or a sizeable block of text (a word, a paragraph or many pages -- size doesn't matter).
- Instead of scrolling through the document to locate the text, the text can be quickly accessed by reference to the bookmark.

- And as used by Pathagoras, retrieval of text, and its insertion into the target document, during a document assembly session is accomplished without even opening the source document. Bookmarks allow smooth and 'silent' transfer of text from source to target.

Creating a Bookmark (the 'basics')

It is easy to add bookmarks to any Word document manually. Pathagoras tools can create bookmarks for you automatically, but you do not need Pathagoras for this purpose. You can do it yourself. Here are the steps:

1. Highlight the text that you want to bookmark.
2. Click the **Insert** button in the Word menu.
3. Click **Bookmark**.
4. At the screen that appears, type the name you want to assign to the highlighted text. (See Name Rules below).

Bookmark Naming Rules:


Whether you are adding new terms to an existing or to a new glossary, or want to change the name of an existing glossary term, there are two bookmark **naming rules** that must be observed:

1. The name must start with a letter of the alphabet or an underscore character. (If you use the underscore character as the first character in a bookmark name, it will make the bookmark 'hidden.')
2. The remaining characters of the name must be letters or numbers or the underscore character. No other symbols, including spaces or punctuation marks, are allowed.

A 'Pathagoras Glossary':

As indicated above, a Pathagoras glossary is a simple Word document containing terms that you wish to use for document assembly purposes, each term of which is bounded by bookmarks to indicate its scope. To enhance the document assembly aspects of the system, however, Pathagoras pre-pends two 'identifiers' immediately above each bookmarked term. This is so you can readily discern the name and subject of the term.

- The first identifier, two lines above the bookmarked text, and in red, is a repeat of the bookmark name. (You could discover the name of the bookmark by clicking within the bookmark itself, pressing Insert|Bookmark. The text in the Name field is the name of the bookmark.) We put the bookmark name on the 'face' of the document to make it easier for you to see the name.

 Please note: the **red line** is simply text. It is not linked to the bookmark. It is just sitting there. Changing the **red line** alone will not change the name of the bookmark. Note also that the color **red** is for emphasis only. The color is not substantively important. (We will cover 'changing the name of a glossary term' in a separate section.)

- The second identifier is the **blue line**. This line will always appear immediately above the bookmarked text. The **blue line** reflects the subject that you assigned to the clause (or if you didn't assign a subject, the one that Pathagoras assigned as a default).

Unlike the **red line**, the **blue line** is somewhat substantive. A change on the blue line will be picked up into your next display of the Clause Selection Screen or in a

DropDown List upon creation or a refresh. Change blue lines with impunity until you have your subjects just the way you want them.



Here is an idea for you to consider when you want to make choices and groupings a bit more 'visible' to the end user. And two (or more) spaces to the subject line when the particular clause is a sub-division of a parent or when you want to group clauses. (For the latter, consider no indent for the first group, a two character indent for the second group, no indent for the third group, etc.)

Changing a glossary term name: As noted above, you cannot change the name of a glossary term simply by changing the **red line**. But you can successfully change the glossary name by following these steps:

1. Place the cursor anywhere within the bookmark area.
2. Click Insert|Bookmark. The name of the bookmark within which the cursor sits will appear at the top.
3. Click the "Go To" button at the bottom of the bookmark screen. This will highlight the entire scope of the bookmark.
4. Click the "Delete" button just to the left of <Go To> (don't worry, only the bookmarks are deleted, not the text).
5. At the top of the screen type the new name. Press <Add> to save the new term.
6. Manually type within the **red line** to reflect the new name. If you wish to modify the subject, make the changes in the blue line.

Changing the scope of a glossary term: If you found a bit of text 'outside' the closing bookmark, don't worry. It is easy to get it inside the closing bookmark. Here are two ways. (1) Cut/copy the 'outside text' (to put it into the clipboard), move the cursor to inside the closing bookmark and paste it in. (2) The second method is to highlight the entire block of text that you want the bookmark to encompass, and click Insert|Bookmark. The name of the first bookmark that the block you highlighted surrounds will appear at the top of the screen. Most likely this is the name you want. Click <Add> and the bookmark will be expanded to include the entire block of text

Adding or deleting new text in a glossary term: Editing text within a bookmark should not worry you at all. A bookmarked section of a document is still just standard text within a standard Word document. Delete, add, copy, cut, paste, text, pictures, tables, number fields. Whatever you want is fully possible. The bookmark ends adjust to your new text without you having to do anything. Just make sure, if you are adding text to the beginning or the end of the term that you are actually within the opening and closing bookends. See section below on "Showing the 'bookends' of a bookmarked term.:

Showing the 'ends' of a bookmarked term: The range of a particular bookmarked term is noted by two light-grey-in-color brackets at the beginning and end of the term. More often than not, they will not be visible without take a few more steps because Word does not display bookmarks by default. Even when 'exposed' the light grey color can be hard to see, so look carefully.

Word provides you one method, and Pathagoras has 2, for you to 'show' the bookends:

- In Word, click Tools|Options|View and check the Bookmarks item.
- In Pathagoras, display either the Main Menu or the Utilities/Settings screen. Look for

and click the button that says **Show Bookmarks** (the buttons are in grey and are at the bottom). When clicked, the button will toggle to **Hide Bookmarks**.

How glossaries work: Even with all of the above in mind, it is not necessarily clear how text gets moved *from* a glossary *to* a new document. Nor is it obvious as to how the names of the glossary terms get displayed into the Clause Selection Screen or DropDown Lists. This section discusses what is happening behind the scenes:

Each time you call on a glossary, Pathagoras opens that glossary to gather information from it.

Taking advantage of the "bookmarks" table that Word automatically creates when a document is saved with bookmarks inside, Pathagoras jumps in a very rapid fashion to each bookmark in the glossary. It captures the name (but not content) of the bookmark and then, for each, jumps back one line to capture the subject line (the [blue line](#)).

Pathagoras then closes the glossary and, with the information in memory, creates the Clause Selection Screen or DropDown List.

When you call for a term in the glossary, Pathagoras quickly opens the document, goes to and copies the text of the specific bookmark, pastes it into the document under assembly. That search, copy, paste process is repeated until all requested terms are inserted.

That is all there is to it!

Debugging a Glossary:

Because a glossary is a plain text document, it is fairly easy to 'debug' it if you are familiar with all of the above 'rules.' However, Pathagoras does not expect that of you, and provides tools to automate this process. If you have issues with a glossary that you cannot explain, you should run a 'Structure & Integrity Check' against the glossary. Here are the steps:

1. Display the glossary onto a Word editing screen.
2. With the glossary 'on display,' click the Pathagoras dropdown features list and select "Editing/Clause Creation Tools".
3. Click "Glossary Tools".
4. Click the <Structure & Integrity> button.

Line endings for glossary terms:

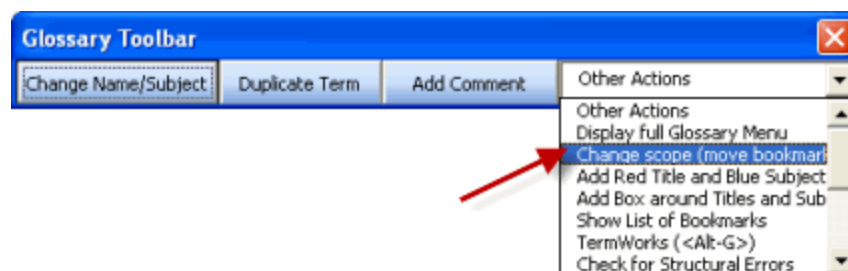
Unlike 'documents' which force you to have a final 'Enter' (pilcrow character) at the end of all text, you are not so constrained with glossaries. You can control whether or not the glossary term contains a final Enter. But failure to exercise this control can lead to some unexpected results, especially when it comes to formatting issues. Read on for the 'why.'

Pathagoras embeds all formatting code for a particular paragraph of text into the 'pilcrow' (The 'pilcrow' is the 'backwards P' shaped character: ¶) character that closes that paragraph.

- If the bookmarks include that final pilcrow, then the text will come in fully formatted.
- If the bookmarks exclude that final pilcrow, the text comes in as unformatted text. The text will take on the style/characteristics of the receiving paragraph.

You can manually change the 'scope' of the bookmarks. To do so, display the glossary. Make sure Show Bookmarks is set to 'on' (see text above for instructions on how to display bookmarks). Study the document, looking especially for whether the final pilcrow is within or without the closing bookmark. To enclose the pilcrow within the closing bookmark when it is not initially, place your cursor just inside the bookmark and press the 'enter' button. Sometimes it is necessary to type a 'real' character (a period will work) and an enter. Then delete the extra character. By playing with the sequence you can accomplish what you want.

You can also use Pathagoras editing tools to accomplish the above. First, click your cursor to somewhere inside the term you want to change. Then, from the Pathagoras dropdown features list, click Authoring/Editing tools. Select Glossary Toolbar (floating). This is what you should see:



From the toolbar that will appear, select 'Change scope'.

Then, choose the action you desire from the next screen that appears:



3.7.4 Glossary vs. Folder of Terms

Which is better: a folder of terms or a glossary?

Pathagoras works equally well whether clauses are stored in a [glossary](#)^[70] or in a [folder of terms](#)^[69]. For most, the folder method of clause storage will initially be the easier and more intuitive. Most users are used to, and very comfortable with, conventional Word folders and documents.

But there are some real advantages to a glossary. Here are the 'big three:'

- **‘Editability’** is a primary one. Consider what you might do if there is a change in a term of art that appears in many of your terms or clauses. Or let’s say you decide to change the name of a variable for one of your books or systems. Or let’s say your business name or your address changes.
 - If you keep your clauses in a folder of individual clauses, you may have to open up all of the documents to effect the change system-wide. (Minimally you would have to open up each document and at least inspect it to see if the change needs to be made.) That could be a lot of work.
 - With a glossary, you just open the one document, and can view the entire document at once. Using “find & replace,” you would then replace the old name/address/reference with the new one. Very quick. Very elegant.
- **‘Transportability’**: Being a single document instead of ‘many documents,’ glossaries are ultimately easier to ‘handle’ in other regards. No special steps are needed to get it from your hard disk to a floppy disk or thumb drive. A glossary is simply a Word document. They are easier to transport and to e-mail (only one file as an attachment, instead of multiple files), and easier to move to another folder.
- **‘Speed’**: Since all clauses are being drawn from the same document (which remains open during the entire assembly process), document assembly using a glossary is slightly faster than requiring Word to open up separate documents. For small documents, the time difference is not noticeable. But we have ‘clocked’ about a 10% time savings (27 seconds vs. 30) for larger documents.

But don’t let the advantages discourage you from doing what you already know. If you think that you will be more comfortable working with documents, don’t elect ‘glossaries.’ Ultimately, Pathagoras does not care. Pathagoras is designed to promote experimentation. If you decide to ‘go folders’ now and to ‘go glossaries’ later, the conversion is easy, and practically automatic.

If at any time you decide that you want to convert a folder of clauses into a glossary, it is an easy process. See [Folder to Glossary](#)³¹⁸.

3.7.5 Adding (‘Shelving’) a Book

Pathagoras provides two distinct ways to add (‘shelve’) a book to the active library.

- If a bookshelf is empty when you are viewing the Libraries & Books screen, you can [Quick Set](#)⁷⁷ the book. Quick Set automates the process of placing a book onto an empty shelf.
- If you wish to replace an existing book with another, or repoint an existing book to a new location (perhaps you moved the files to a server or to a cloud location), hold down the Ctrl- button while clicking on the book you wish to replace or repoint. [This page](#)⁷⁸ discussed repointing in more detail.
- If you wish to rearrange the books in a particular library, do so [via the Settings screen](#).⁷⁸

3.7.5.1 Quick Set

To quickly assign any existing Word folder as a document assembly source:

1. Click the Document Assembly icon:



Figure 1. The Pathagoras Toolbar
(left=Word 2003 and prior; right= Word 2007 and later).

The **Libraries & Books** screen appears (Figure 2, below).

2. Click on an empty bookshelf. (In the below example, bookshelf 7 is empty. A 'Quick Set' button is revealed in the lower right side of the screen.)
3. Click it.

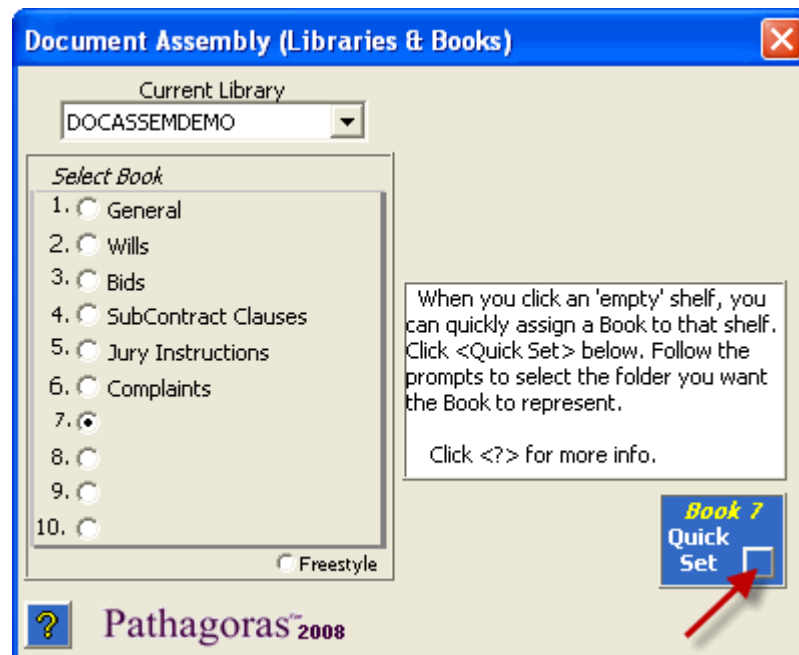


Figure 2. Libraries & Books screen.
The <Quick Set> button is revealed when an empty bookshelf is selected.

4. Pathagoras displays a menu from which you can choose the 'type' of book to place on the shelf. Examine the choices. For this example, select "DOCUMENTS in a FOLDER." (This will be choice #1 or #2.)

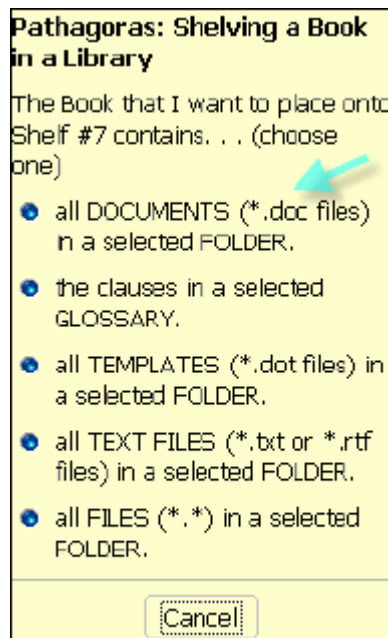


Figure 3. The 'Select Type' screen.
Choose the content of the book to place on the empty shelf

5. Follow the remaining prompts and navigate to the folder (any folder) that you wish to be a source for document assembly clauses. If you are just getting started, just select a folder that contains documents that you frequently use.
6. When you have navigated to the desired folder, select one document in the folder.

→NOTE: *It absolutely does not matter which document you select. The selection is only to confirm that you have finished navigating, that you are inside of the desired folder, and so that the OK button can become 'hot.'*

7. Press the "OK" button in the window to lock in the folder choice. The remaining prompts allow you to assign a nickname to the folder, and to make certain other 'administrative' decisions. When done, close any residual overlay screens and return to a blank page.

NOTE: The Quick Set process described above only works when a bookshelf is blank. To replace an *existing* book, you must go through the Settings screen. That is the topic of the [next section](#) ⁷⁸.

3.7.5.2 Pointing/Repointing via Settings Screen

As your system matures, you may decide to move your source documents. Perhaps you have moved them to more logical locations, or have refined the scope of the content of the folders. Perhaps everything is relatively intact, but you may have moved your files and folders to a server or to the cloud for everyone in the office to use. (See more regarding moving files and reorganizing your systems [at this link](#) ⁶⁵⁴.)

If you have initially pointed one of your books to location 'A' but they are now in location 'B', you will need to repoint the book to comport with the new file locations. (Remember, a 'book' is merely a pointer to a folder. So all you are doing with 'repointing' is telling

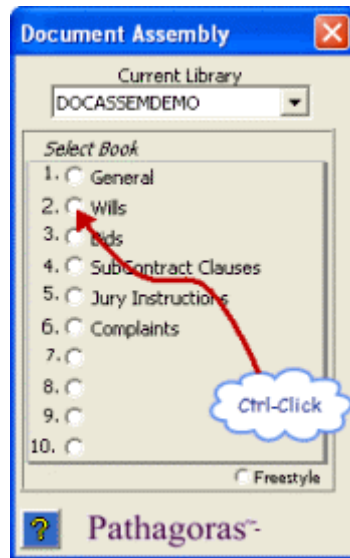
Pathagoras the new location of the folder.)

Here are the steps:

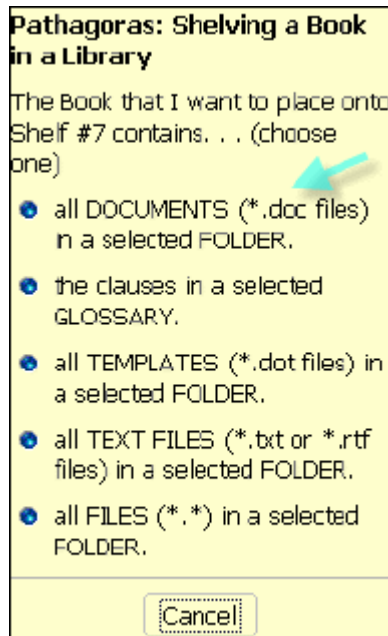
1. Click the Document Assembly button (Figure 1) and display the library that contains the book you want to move.



Figure 1. The Pathagoras Toolbar
(left=Word 2003 and prior; right= Word 2007 and later).



2. **Ctrl-click** on the book you want to move. A menu will appear from which you can choose the 'type' of book to place on the shelf. Examine the choices. For this example, select "DOCUMENTS in a FOLDER." (This will be choice #1 or #2.)



Choose the content of the book to place on the empty shelf

4. Follow the remaining prompts and navigate to the folder (any folder) that you wish to be a source for document assembly clauses. If you are just getting started, just select a folder that contains documents that you frequently use.
5. When you have navigated to the desired folder, select one document in the folder.

➔ **NOTE:** *It absolutely does not matter which document you select. The selection is only to confirm that you have finished navigating, that you are inside of the desired folder, and so that the OK button can become 'hot.'*

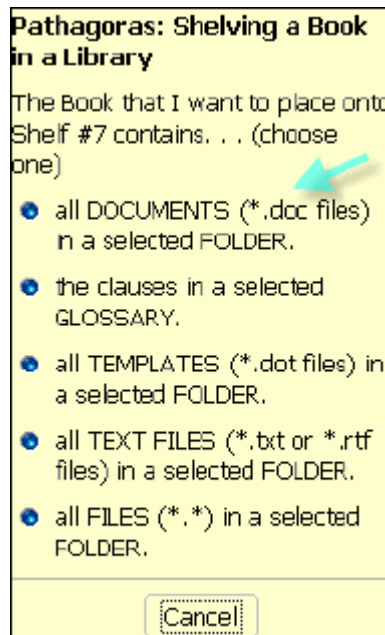
6. Press the "OK" button in the window to lock in the folder choice. The remaining prompts allow you to assign a nickname to the folder, and to make certain other 'administrative' decisions. When done, close any residual overlay screens and return to a blank page.

Alternative.

The below is an alternative approach to repointing a book. It takes more steps, and is not the 'preferred' method, but it does let you explore the Document Assembly Settings screen, in case that interests you.

1. Click the Document Assembly button and display the library that contains the book you want to move.
2. Display the Document Assembly Settings screen (Two methods are described in the link.)
2. Click the bookshelf on which you want the new book to appear (or the one you wish to repoint). Once you select a shelf, two new buttons will appear toward the bottom of the screen. One says 'Browse' and the other says 'AutoSet'. Click <Browse>.

3. Pathagoras displays a menu from which you can choose the 'type' of book to place on the shelf. Examine the choices. For this example, select "DOCUMENTS in a FOLDER." (This will be choice #1 or #2.)



Choose the content of the book to place on the empty shelf

4. Follow the remaining prompts and navigate to the folder (any folder) that you wish to be a source for document assembly clauses. If you are just getting started, just select a folder that contains documents that you frequently use.

5. When you have navigated to the desired folder, select one document in the folder.

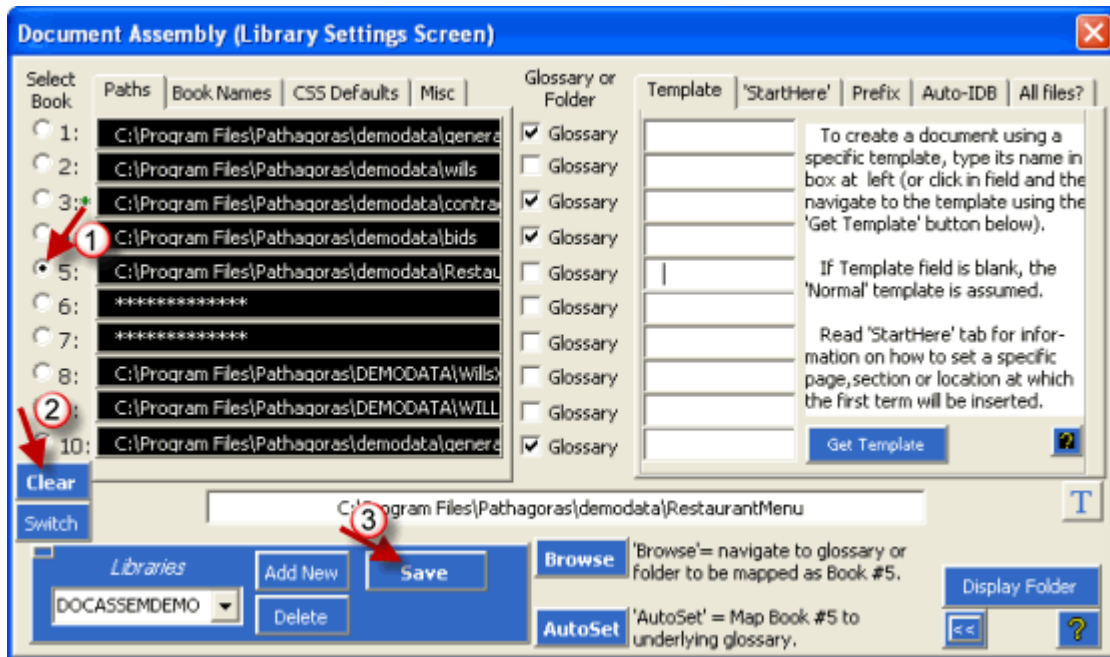
→NOTE: It absolutely does not matter which document you select. The selection is only to confirm that you have finished navigating, that you are inside of the desired folder, and so that the OK button can become 'hot.'

6. Press the "OK" button in the window to lock in the folder choice. The remaining prompts allow you to assign a nickname to the folder, and to make certain other 'administrative' decisions. When done, close any residual overlay screens and return to a blank page.

3.7.6 Removing ('unshelving') a Book

To remove a book from a shelf:

1. Click the Document Assembly icon. The library from which you wish to clear or reassign a book should be active.
2. Click the book you wish to remove. When the screen expands, check the "Settings" box just above the Next button. Click Next and the Document Assembly Settings screen will appear:



3. Click on the option circle adjacent to the book's name (1). When you do so, a <Clear> button will appear at the bottom of the column of book name(2). Click <Clear> to remove the book from the library.

4. Press the <Save> button (3) to confirm the action.

➔ **NOTE:** A book is simply a *pointer* to the folder or glossary that the book represents. The book is not the actual folder nor is it the actual files in the folder or the glossary. Deleting a book removes only the pointer for Pathagoras purposes. It will not affect the actual files.

See also: [Deleting a Library](#) ⁸⁵

3.8 Creating a New Library

You should create a new library when you are ready to organize your form documents (the ones that you will use for document assembly) into more logical groupings. You should also create a new library when the current library is 'full.'

- For example, in a law office, you may wish to have a separate library for each major practice areas. The books within each library would contain both boilerplate and boutique text needed to build documents on a specific topic.
 - You might first create an Estate Planning library, populated with books of clauses on the following topics: Wills, Trusts, Powers of Attorney, etc.
 - You might create a second library called "Contracts." Books in that library might be titled Real Estate (Commercial); Real Estate (Residential); Purchase Agreements, etc.. Pour appropriate clauses into each book.
- There are no 'right' or 'wrong' configurations. Your office needs will drive the assignments of libraries and the books placed inside of them.

To create a new library:

1. Click the Document Assembly icon:



Figure 1. The Pathagoras Toolbar
(left=Word 2003 and prior; right= Word 2007 and later).

The Libraries & Books screen appears.

2. Drop down the list (near top of screen) of available Libraries.:

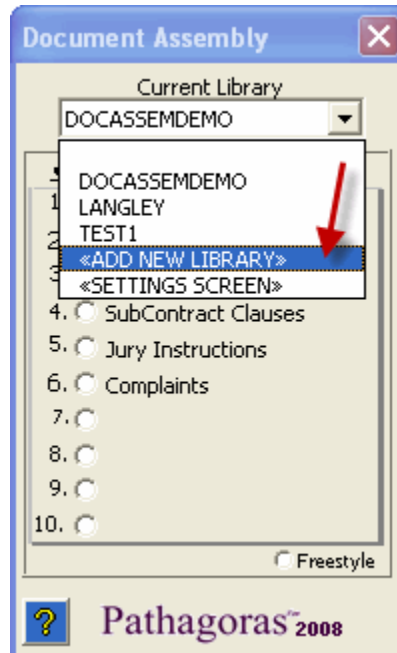
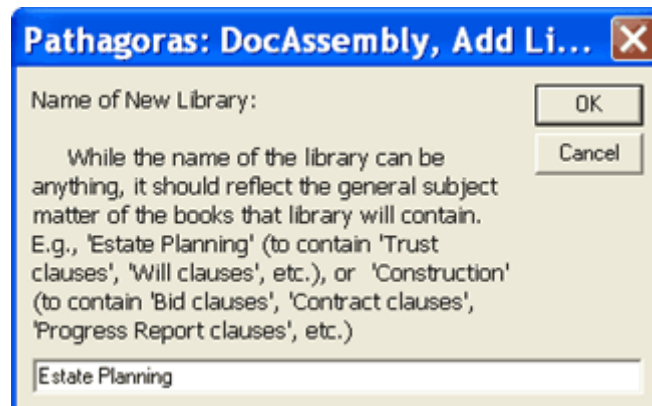


Figure 2. Initial Document Assembly Screen: "Libraries & Books"

3. Select <<Add New Library>>. The following screen will appear:



4. Provide a name for your new library in the screen. After you click OK, the Document Assembly 'Settings' screen will appear, showing the previous Default Library's settings (Figure 3). It will then lay over that a screen (Figure 4) asking how much, if

any, of the previous Library's settings you wish to keep.

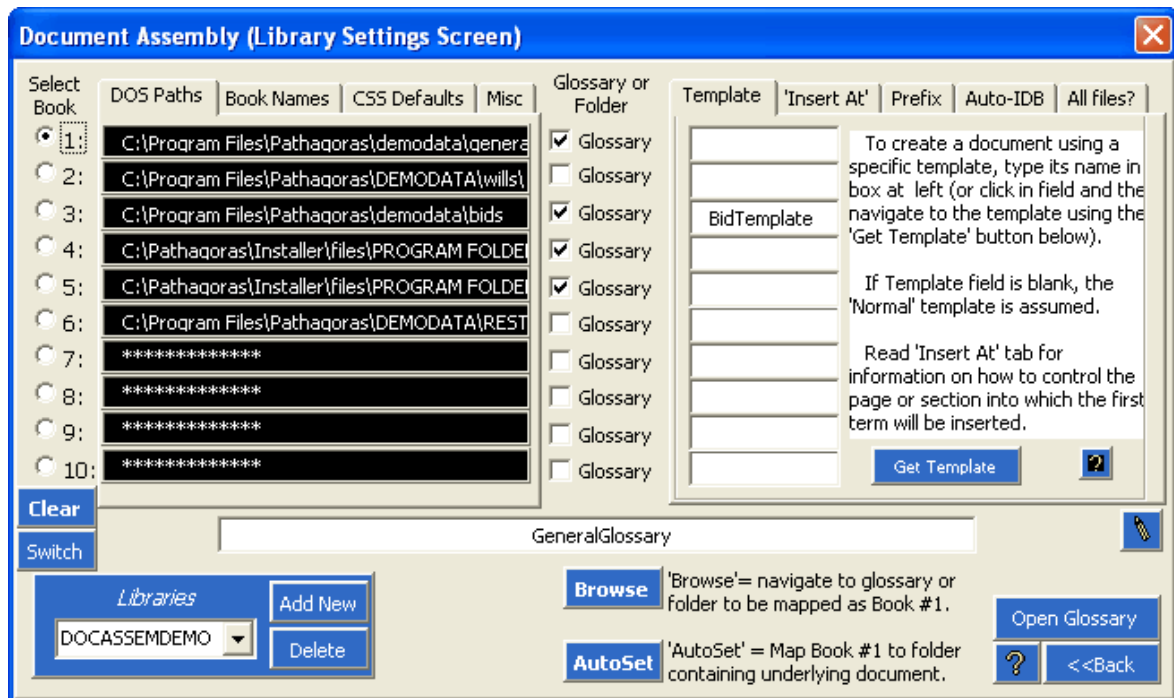


Figure 3. Document Assembly Settings screen

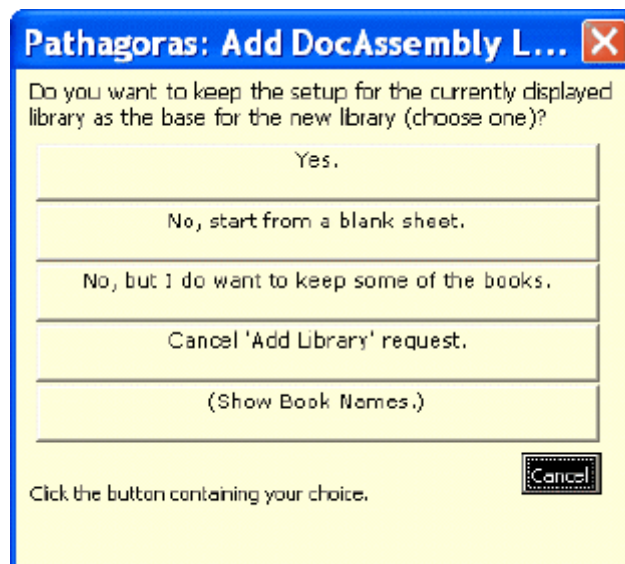


Figure 4. Let Pathagoras know which elements of the 'old' Library's Settings (if any) you want to carry over to your new Library

5. Pathagoras now wants to know how much of the current library's books you wish to preserve. (Figure 5) Click the top button if you want to preserve the current setup. (If you preserve the current setup, you can change any setting later.) If you want none of them, click the second entry and your new library will be empty. For this example, click the second button down and let's start 'from a blank sheet.'

You now have (at least) two libraries: (1) the original DocAssemDemo library, and (2)

the one you just created plus any libraries you created earlier. The one you last created will be the new 'default' library.

You can see all libraries, and choose among them, by clicking the drop-down list under the words "Current Library" in the Libraries & Books screen (Figure 2 above). The last library selected will become the new 'default' library.

Alternative method: You can add a new Library directly from the Settings screen. Click the Settings button in the lower right corner of the Document Assembly screen. Then, when the Settings screen appears, click the Add New button and follow the prompts.

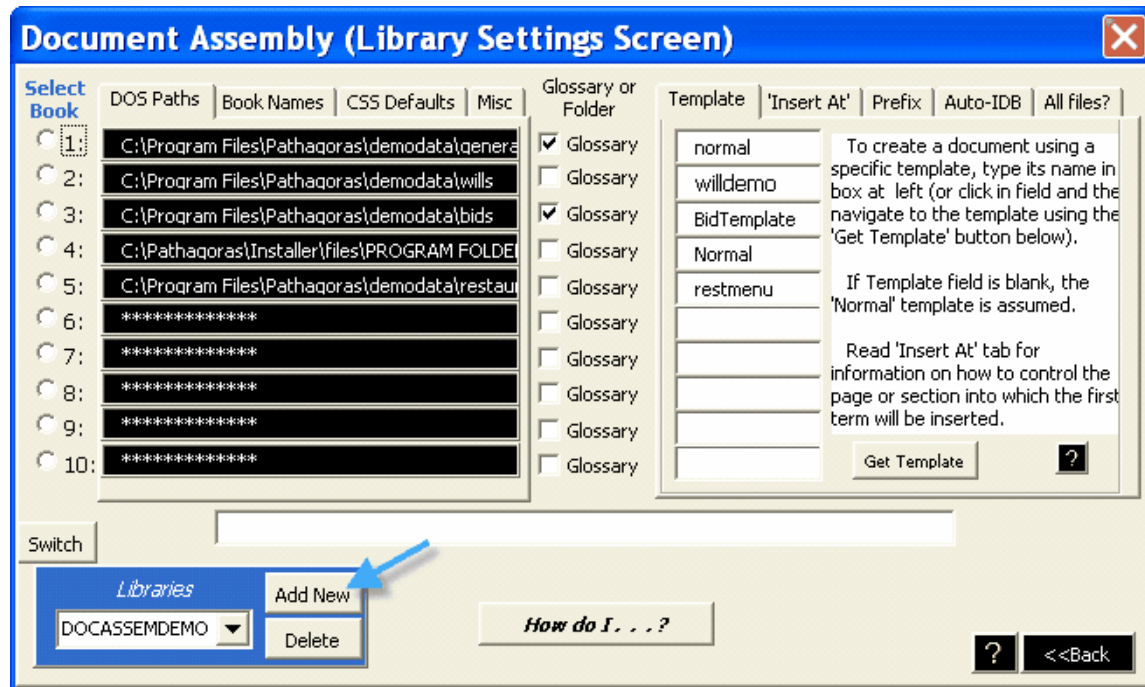


Figure 5. Instead of adding a new library from the initial Document Assembly ('Libraries & Books') screen, you can activate the Setting screen and click the 'Add New' button there. The remaining steps are identical.

3.9 Deleting a Library

To delete an entire library, or remove a single book from one of its shelves.

1. Activate the Document Assembly Settings screen.
2. Display the library you wish to delete (or from which you wish to clear or reassign a book).
3. To delete the entire library, click the Delete button in the lower left side of the Setting screen. A series of confirmation screens will appear to insure that you do not accidentally delete a library.
4. To remove a single book from a shelf, click on the option circle adjacent to the book's name. A <Clear> button will appear. Click it to remove the book from the library. Press the <Save> button to confirm the action.)

➔ A library, and the books shelved within the library, are pointers to the folders or glossaries that the books represent. They are not the actual folders or the files in the folders

or the glossaries themselves. Deleting a library or a book does not affect any substantive files.

3.10 Settings Screen

The Settings screen is the heart of the document assembly system. Click the Settings button on the Libraries & Books screen.

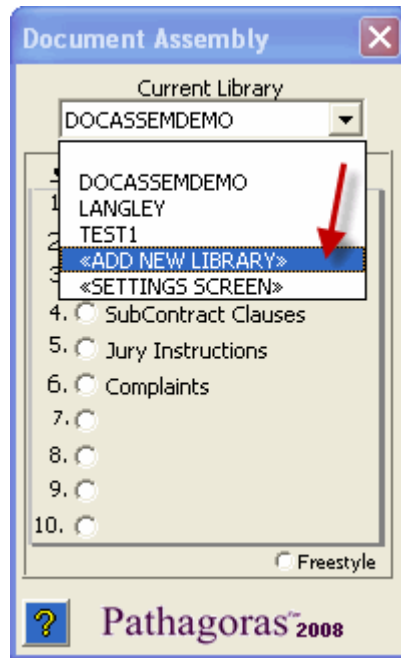


Figure 1. The Libraries & Books screen. Click <Settings>.

Click the Settings button. From there, you

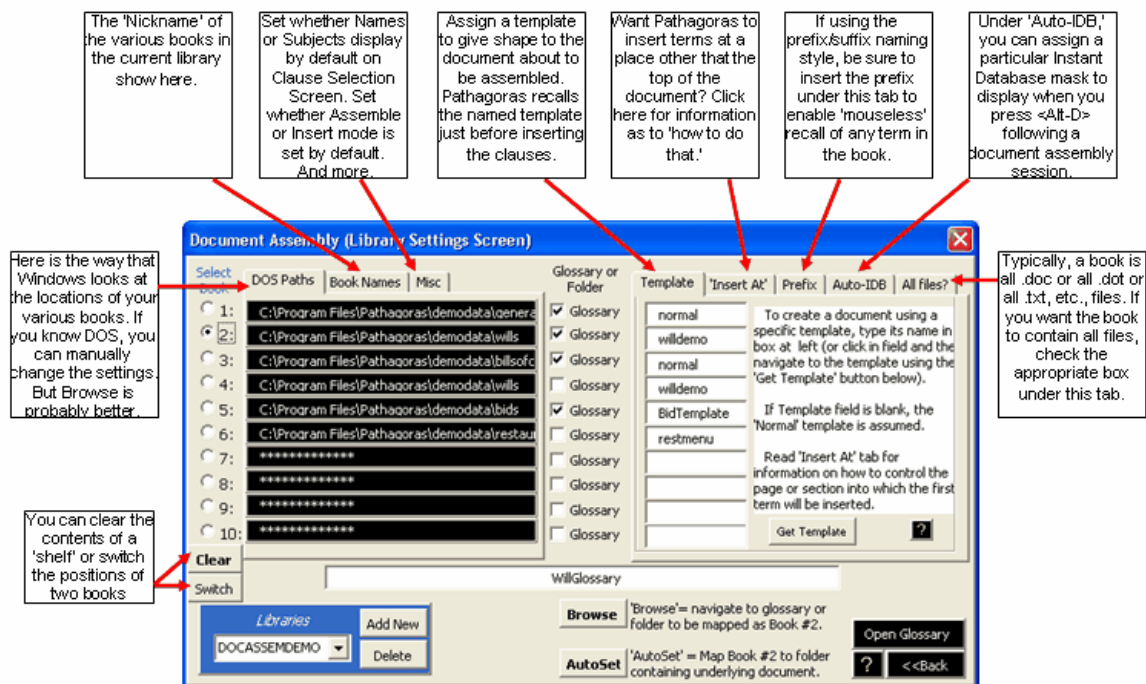
- can place a book on a shelf. See Adding a Book to a Library
- assign a template to serve as the 'shape' and form upon which to assemble a particular book.

See: [Templates](#)⁹⁶, [Assigning a Template to a book](#)¹⁰¹

- assign a prefix (i.e., link a particular folder or glossary to the prefix by which the clauses may be named).

See Prefix/Suffix Naming Convention, Assigning a Prefix to a book.

- and more. The below screens illustrate the many tasks that can be accomplished via the Settings screen.



3.10.1 Assigning a Template

Documents will typically (and by default) be assembled into a new document using the "normal" template as the base. All styles, margins, headers and footers associated with the normal template will be active. If you have a special template on which you wish a certain book to be assembled, note the name of the template in the Settings screen, under the templates tab.

3.10.2 Assigning an IDB Mask

You can pre-assign the Instant Database mask that will display when you press <Alt-D> after a document has been assembled. From the Document Assembly Setting screen, click the Auto-IDB tab. Select the appropriate mask from the dropdown list.

See Also:

[Instant Database](#) 186

[Masks](#) 260

[Assign Mask to Book](#) 269

3.10.3 Assigning a Prefix

If you have named the terms of a particular book using the Prefix/Suffix Naming Convention, and have shelved that book into the current library, you can create a permanent association between the prefix and the folder or glossary that houses the clauses using this

tool.

Click the Prefix tab and, on the appropriate line, type the 2, 3 or 4 character prefix. Then click the <Assign> button.



If the prefix you want to assign is not associated with any of the currently displayed books, you can either:

- recall a different library (using the dropdown in the lower left corner of the Settings Screen) or
- click the Prefix Table button and manually register the prefix to a designated folder or glossary.


See Also:

[Prefix/Suffix Naming Convention](#) ⁴²⁰

[Prefix Table](#) ⁴²⁴

3.10.4 Show All Files in Folder

By default, Pathagoras will filter out all 'non-document' files in the designated folder. If you wish to always display every item in the selected book, check the appropriate box found under the 'All Files?' tab.

 If sometimes you want to display all files but sometimes you don't, leave the 'All Files?' checkbox unchecked. When you are in a document assembly session, selected the desired book that you want to display and click Next. The 'normally' filtered display will be presented in the Clause Selection Screen. Now, click the Show All button toward the lower left side of the screen. The left ("Available Files") panel will be repainted to show all files in the folder.

3.11 SuperBooks

Imagine being able to recall any document stored in a designated location just by typing its name onto your editing screen and pressing a hot-key. Pathagoras lets you do just that. Designate any folder on your system (local, LAN or WAN) as the **SuperFolder**, and/or any glossary as the **SuperGlossary**, and you are all set.

Limitations: Really, none.

If you are *not* typing the document name at the left margin, the document you desire must be a 'single word' names to the documents that you store in your SuperFolder and wish to recall with the Alt-G method.

"My_Long_Name_Document" is considered a single word, as is
MyLongNameDocument.

But if you are typing the document name beginning at the left margin, the document name can include spaces.

"My Long Name Document" is just fine.



When it comes to a SuperFolder, please remember that you can direct Pathagoras to look not only in the designated folder, but in the first layer of sub-folders beneath the designated folder. This will provide you the ability to store a tremendous quantity of documents, yet keep them organized in appropriate categories.

See these pages for specific setup instructions and additional information:

[SuperFolder](#)  90

[SuperGlossary](#)  91

[Order of Search](#)  95

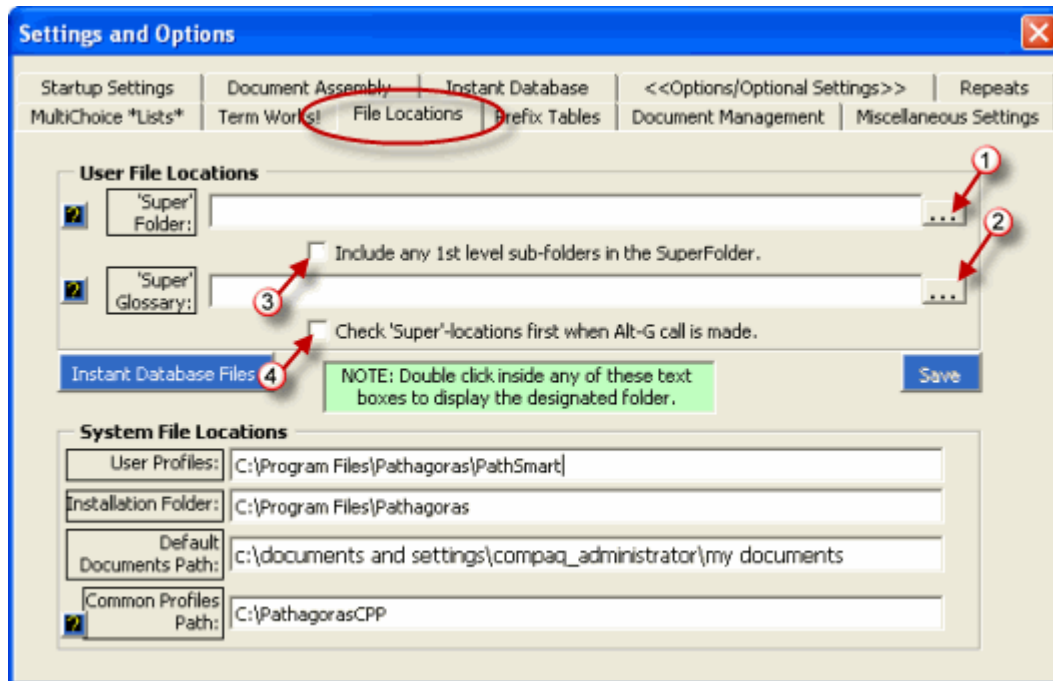


Consider storing 'classic' forms, form letters, boiler plate text, etc., in a SuperFolder. Use the SuperGlossary to store what are likely the more numerous items such as signature blocks, client, customer, and patient telephone numbers and addresses. (Remember -- a glossary is a single document which can contain hundreds, even thousands, of individual clauses. Perfect for the kinds of items just listed.)

3.11.1 SuperFolder

Designating the 'SuperFolder'

- (1) From the Pathagoras features menu, click the Utilities/Settings button.
- (2) Click the <All Settings> button. When the All Settings screen appears, click the "File Locations" tab.



(3) Click the 'Set' button to the right of the box labeled SuperFolder (label 1 in the figure).

(4) Navigate to and select one document within the folder you want to designate as the SuperFolder (It does not matter which item you select. The selection is only to confirm that you have finished navigating.) Press the "OK" button in the window to lock in the folder choice.

(5) Alternatively, you may type (or paste in) the location of the folder. A "Save" button will appear at the right. Click it to lock in the designation.

(6) If you wish Pathagoras to search for documents within the top level and the first level of sub-folders in the SuperFolder, click the box labeled "3" in the figure above.

(6) If most of your <Alt-G> calls will be to the SuperFolder, you might considering 'promoting' the SuperFolder to the top position in the [Search Order](#)⁹⁵. Click the box labeled "4" in the figure above.

See also:

[SuperGlossary](#)⁹¹

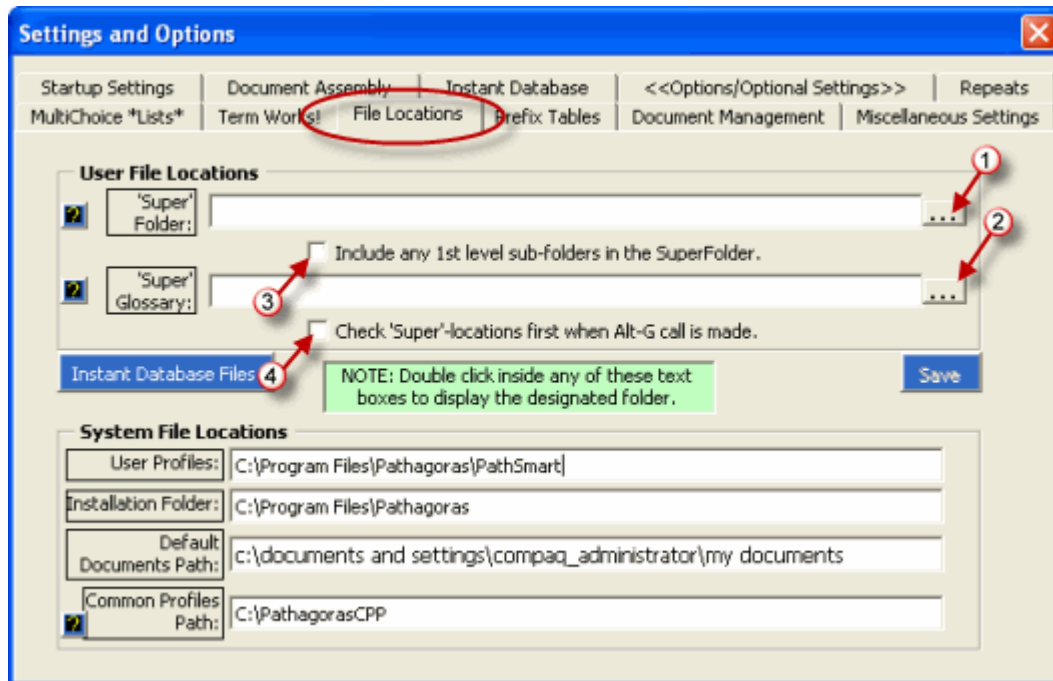
[Order of Search](#)⁹⁵

3.11.2 SuperGlossary

Designating the SuperGlossary

(1) From the Pathagoras features menu, click the Utilities/Settings button.

(2) Click the <All Settings> button. When the All Settings screen appears, click the "File Locations" tab.



(3) Click the 'Set' button to the right of the box labeled SuperGlossary (label 2 in the figure).

(4) Navigate to and select the glossary you want to designate as your SuperGlossary. Press the "Open" button in the window to lock in the choice.

(5) Alternatively, you may type (or paste in) the full path and name of the glossary. A "Save" button will appear at the right. Click it to lock in the designation.

(6) If most of your <Alt-G> calls will be to the SuperGlossary, you might considering 'promoting' the SuperGlossary to the top position in the [Search Order](#)⁹⁵. Click the box labeled "4" in the figure above.

See also:

[SuperFolder](#)⁹⁰

[Order of Search](#)⁹⁵

3.11.3 Adding Terms to a SuperBook

Adding terms automatically: To add a new term to a SuperBook, highlight the term and press <Alt-G> ('Highlight & Add'). The **Term Works!** screen will appear. Provide a name for the new term (a single word name if you plan to add the term to the SuperGlossary) and give it a more descriptive subject. Press the <Other Book> button near the center of the **Term Works!** screen. This will take you to the listing of the available 'target' glossaries and folders. Select the SuperFolder or the SuperGlossary checkbox at the bottom left of the screen.

Adding terms manually: You can manually add any document to your SuperFolder by copying or moving or otherwise saving the document in the folder that you designated as your SuperFolder. No 'registration' is required. Its 'being there' is all that is needed. (Remember, with SuperBooks and most other Pathagoras features, there are no special tricks

or complex rules in play.)

You can also place a new (or existing) document into your SuperFolder using [SaveSmart](#)⁵⁴⁰. Once designated, your SuperFolder is assigned as SmartPath 13. It will be listed with your other SmartPaths on the SaveSmart screen. (Click [here](#)⁹⁰ for instructions on how to designate a SuperFolder.)



While you can name documents stored in your SuperFolder just about anything you wish, you should consider naming those documents with easy to recall or easy to determine names. A primary purpose of the SuperFolder is to act as a repository for documents that can be recalled simply by typing their names to the screen and pressing <Alt-G>. If those documents are saved with complex or multi-part names, you likely will not be able to readily recall their names to retrieve them.

See Also: [Order of Search Rules](#)⁹⁵

3.11.4 Compare to Prefix/Suffix Naming

The prefix/suffix naming convention is an extremely powerful tool. Name your clauses with a prefix containing two, three or four letters followed by a suffix of at least three number, and you can then instantly recall any clause so named from wherever its location. You simply type its name and press <Alt-G>. Read more about this powerful tool under [Prefix/Suffix Naming Convention](#)⁴²⁰ and [<Alt-G>](#)⁹⁴.

The 'limitation' of the prefix/suffix naming style is that a special naming convention must be followed. It presupposes that the clauses are 'categorizable' (i.e., it must fit into the subject matter that the prefix represents; e.g., 'will' for 'wills'; 'con' for contracts).

SuperBooks have no specific naming rules. SuperBooks are intended to contain those clauses that span all subjects and are used in every context. Therefore, you should consider pouring all of your 'general' clauses, terms and forms (form letters, signature blocks, lists of addresses and phone numbers, office boilerplate text, etc.), into one or the other of your SuperBook.

3.11.5 Compare to AutoCorrect/AutoText

The experienced user will observe that a SuperBook has many similarities with Word's built in AutoCorrect and AutoText functions. The main benefit to Pathagoras' version is that it is universal.

- Unlike AutoCorrect (which is typically computer bound), everyone on the network can easily point to the same SuperFolder and SuperGlossary.
- There is no limit to the type of information you can store in a SuperBook. Images, tables, highly formatted text. A SuperBook can handle it all.
- Since a SuperBook is just a Word document, you can take any component of it home with you to edit and refine it, just like any other Word document. (With Autocorrect, you have to be on the same computer, recall the autocorrect term, edit it and then resave it. Pathagoras' approach is much simpler because you are always working with a document.)

- Pathagoras does not dissuade the use of AutoCorrect. It is a wonderful tool for correcting your misspelling and completing words such as days of the week and months of the year. However, when it comes to storing universal large blocks of text, consider the SuperBook. (Use Pathagoras to complement Word's offerings, not to supplant them.)

3.12 Mouseless Assembly

Mouseless assembly using <Alt-G>

The "G" stands for "Get". When pressed, it will get a wide variety of different items or functions for you, depending upon the context in which it is called. Here is a summary.

When <Alt-G> is pressed next to *unhighlighted* text:

- if pressed against a number, it will open the [SmartPath](#)⁵³⁶ assigned to that number in the current profile.
- if pressed against a word, it will assume that the word is either a document or a clause name and will attempt to locate that name in the following locations in the order listed:
 - If the term is 'fully qualified,' Pathagoras will look in the designated location and no further. ('Fully qualified' means that the document is pointed to by drive, folder, subfolders(s) and name. E.g., "c:\office forms\estate planning\trusts\living trust clauses\rlt134.doc".)
 - If Pathagoras sees only a term name, it will determine if the term name meets the prefix/suffix naming criteria (two to four letters followed by three or four numbers). If it does, it will look in the book associated with the prefix in the [Prefix Table](#)⁴²⁴.
 - If not found in the above locations, or if the term is not a prefixed name, Pathagoras will look in the **Position #1 book** (The 'Position 1' book is the book that occupies the first bookshelf in the Current Library.)
 - If not yet found, Pathagoras will look in the [SuperFolder](#)⁹⁰ (if designated).
 - If still not found, Pathagoras will look in the [SuperGlossary](#)⁹¹ (if designated).

(You can tell Pathagoras to look in a **SuperBook** before the **Position #1 Book**. Check the box found in Utilities/Settings/File Locations labeled "Check SuperBooks" first.)

- If still not found, Pathagoras will look for the term in your Auto-Correct dictionary.

See Also: ["Order of Search"](#)⁹⁵.

When <Alt-G> is pressed next to *highlighted* text:

- Pathagoras will presume that you want to *add* the highlighted text to one of your books. <Alt-G> will call up the **Term Works!** screen. You can then give the highlighted text a name and an appropriate subject. Then you can select into which book of the current library you wish to save the highlighted text.

3.13 Order of Search

Pathagoras has a definite 'order of search' pattern of which you should be aware to maximize efficiency in naming and recalling your clauses. When a term is recalled (either from a Clause Set or via Alt-G) Pathagoras will look in the following locations in the order listed (this is also called the 'Hunt Path'):

- **If the term is 'fully qualified,'** Pathagoras will look in the designated location and no further. ('Fully qualified' means that the document is pointed to by drive, folder, sub-folders(s) and name. E.g., "c:\office forms\estate planning\trusts\living trust clauses \rlt134.doc".)
- If not fully qualified, Pathagoras will first look in the folder from which the base document is located. (Applicable when base document was called via the Document Assembly routine or from a DropDown List.)
- If Pathagoras sees only a 5 to 8 character term name, it will determine if the term name follows the ['prefix/suffix naming'](#)⁴²⁰ criteria (two to four letters followed by three or four numbers). If it does, it will look in the **book associated with the prefix** in the Prefix Table.
- If the text is a number from 1 to 12, Pathagoras will open the folder paired with the number in your current profile in PathSmart.
- If the text is a name stored in your QuickLinks list, Pathagoras will open the folder paired with the name stored in your QuickLinks list.
- Pathagoras will then look to see if the term name exists in the ['Tag Along'](#)¹¹² section of the document. (Here, the desired text must have been previously 'bookmarked' with the term name.)
- If not found in the above locations, or if the term is not a prefixed name, Pathagoras will look in the **Position #1 book** (The 'Position 1' book is the book that occupies the first bookshelf in the Current Library.)
- If not yet found, Pathagoras will look in the [SuperFolder](#)⁹⁰ (if designated).
- If still not found, Pathagoras will look in the [SuperGlossary](#)⁹¹ (if designated).
(You can tell Pathagoras to look in a **SuperBook** before the **Position #1 Book**. Check the box found in Utilities/Settings/File Locations labeled "Check SuperBooks" first.)
- If still not found, Pathagoras will look for the clause in each active DropDown List.
- If still not found, Pathagoras will look for the term in your Auto-Correct dictionary (Microsoft Word feature).
- If still not found, Pathagoras will look in the first 15 glossaries that you may have stored in your SuperFolder. (This last search presumes that you have not earlier looked in these glossaries via the prefix search.)



To avoid redundant searches (and to speed up the search process), do not store the following items in your SuperFolder:

- (1) your SuperGlossary or

(2) any glossary or any term which you have associated with a prefix.

See also:

[Prefix/Suffix Naming Convention](#)  420

[Prefix Registration Table](#)  424

[SuperBooks](#)  90

[Tag Along clauses](#)  112

3.14 Templates

Document assembly 'jargon' uses the word 'template' in two overlapping but nevertheless distinct fashions.

- A 'template' is an essentially 'blank' document which has been specially constructed to provide the shape (margins, fonts, styles, headers, footers, etc.) to a future document that will be built upon it.
- A 'template' is a complete document containing, in addition to the 'shape' elements described above, the entire body of a typical document type. A 'contract' template, a 'will' template, a 'fax' template, etc

This is the definition attributed to 'template' by most other document assembly programs. Such a template contains every term that possibly might be included in the final document, and then some. By answering questions that you have to build into the template, or using an ancillary file, you pare away the text in the template that doesn't fit the need.

Pathagoras defines 'template' in a more classic sense:

- A Pathagoras template is an ordinary Word '.dot' file that provides the basic shape to the documents that are built upon it.
- A Pathagoras template provides the styles, fonts, headers and footers, the background colors, etc., of the document that is built upon it.
- A Pathagoras template is typically void of any substantive text. However, introductory text, letterheads, page numbers and the like are frequently part of the template.

This is the logic and flow implemented by Pathagoras when you click the Next>> button after selecting one or more clauses from the Clause Selection Screen:

(1) Pathagoras checks to see if a template was associated with a particular book from which the clauses were selected. If so, that template is called and 'laid out' as the receiving document. If no template is associated with the book, the template associated with the first document assembled is 'laid out.'

(2) The clauses selected in the Clause Selection Screen are then quickly 'poured' into that template.

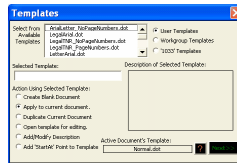
Pathagoras allows you easy access to the various templates you might use in the course of

your document assembly day, and allows you to make various settings to those templates which promote the ease and speed by which your projects can be assembled.

3.14.1 The Templates Screen

As explained in the previous section, a classic Pathagoras template is not a complete (i.e., "overbuilt") document from which you pare away clauses not needed for the particular customer or client. A template is essentially a blank document that provides the shape for the clauses that will be poured into it. The templates screen gives you quick access to the various 'shape' providing devices. (We are talking not just about memo forms and letter formats that Word makes available to when you click "New|Document." Rather, they would include your unique templates designed for your major document production projects.

Activate the 'Templates' screen by dropping down the Pathagoras features menu. Click "Templates." This is what you will see:



The Templates Screen ('Apply' is pre-selected by default)

From the list at the top of the window, choose the template you wish to apply to the currently visible document. The templates screen allows you quick access to any of 3 folders of templates.

Then, select an action you want performed with the selected template.

Useful Information about templates and the template screen:

- If the document name of the template is not descriptive enough for you, you can add a description that provides more information as to its use. Just click the 'Add/Modify Description' and then type/modify the description of the template in the 'Description' block. Press <Save> when done.
- As indicated at the beginning, Pathagoras does not define templates in the same way that other programs do. Templates provide shape. Everything else provides the content. The consequence (all positive) is that what you may have considered a 'requirement' (that all 'source documents' be built as and in ".dot" files) is simply incorrect. Indeed we strongly recommend that all source documents be designed and saved as standard Word '.doc' (document) files. '.doc' files are easier to work with, easier to save and easier to retrieve. (Have you ever tried to save a new .dot files? Word always jumps you to the default templates path, and you have to re-navigate to the folder in which you really want the file.)
- Pathagoras allows you to go 'round-robin' with the templates that display in Pathagoras' Template's screen. That is, it will apply each template against the current document to see how your text looks inside of the template's settings. (This is similar to what Word2007/2010 now does with its styles feature on its Home page.) As you might imagine, this round-robin technique only works well if the template otherwise has no 'main story' text within it.

- The template 'round-robin' is particularly helpful in the following situation: Let us say you draft a letter that you anticipate printing onto your quality letterhead. In that situation, you only want the body of the letter to appear on screen. Now you decide that after you print on the letterhead, you now wish to e-mail or fax it. Of course, you could scan the letterhead version (that takes time). The better technique would be to 'apply' your letterhead template (using the Templates screen) and then e-mail or fax the letter directly from your computer.

3.14.2 Creating a Template

A 'blank' template is in the nature of a 'clear canvas' onto which clauses selected for assembly can be poured during a document assembly session. Keep in mind that while the canvas is 'blank,' it still contains the desired document 'shape' (margins, styles, headers, footers, etc.) and styles needed to give the assemble document the desired look.

When you are having style issues (that is, the style of the text which you are trying to copy/paste into a new document don't appear identical -- see this link for more: [styles](#)³²²)

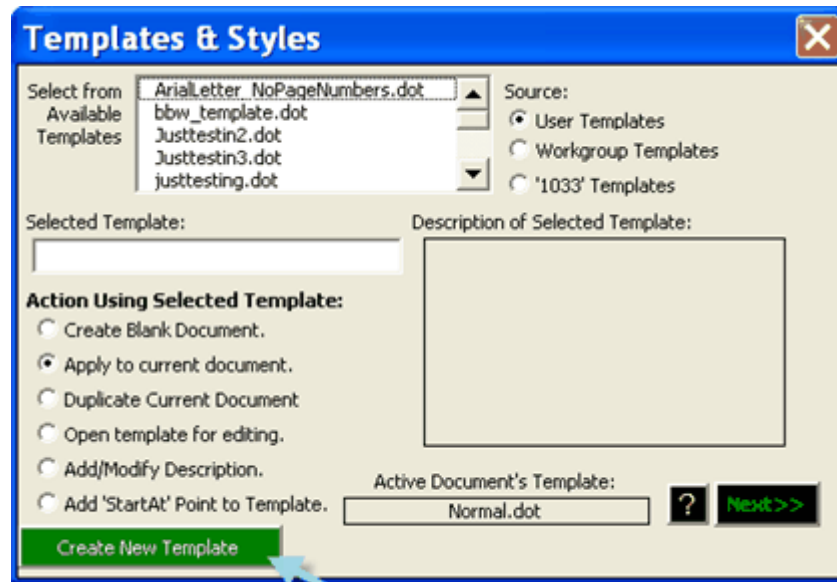
Here is how to create a 'blank' template

Manually

- (1) Display a source document that has the desired 'shape'.
- (2) Press Ctrl-A to highlight all of the text.
- (3) Press Delete to void the document of all text. (Don't panic, we will not lose your original, but make sure that you press **SaveAs** in the next step.
- (4) Save this blank document out as a template. Do so by pressing 'SaveAs' from Word's 'File' toolbar item and then choose Template from the "Save as type:" list at the bottom of the screen. Name the template something meaningful (like 'MyCompany.dot') and save it in your default templates folder. (In most cases, Word will automatically switch to the proper folder when you click on Templates as the Save as type:") Close the template you just created.

Automatically:

- (1) Display a source document that has the desired 'shape'.
- (2) Display the Templates and Styles screen. (It is found in the Pathagoras drop down features menu.) This is what you will see:



(3) Click the <Create New Template> button.

(4) Follow the prompts. (The steps outlined above in the 'Manually' section will be performed, but automatically so.

Now, and from now on, when you assemble a document from the book to which have attached the template, Pathagoras will first lay down the assigned template. (Look at this as the 'canvas' upon which the document will be painted.), Then, all clauses you selected from the Clause Selection Screen will be poured onto the canvas.

Headers and Footers within your new template:

Headers and footer will remain in the template, even if the document otherwise appears blank. If the template you created had 'personalized' headers and footers (i.e., they contained the name of a specific client or customer), don't forget to replace the personal names with the appropriate bracketed variable.

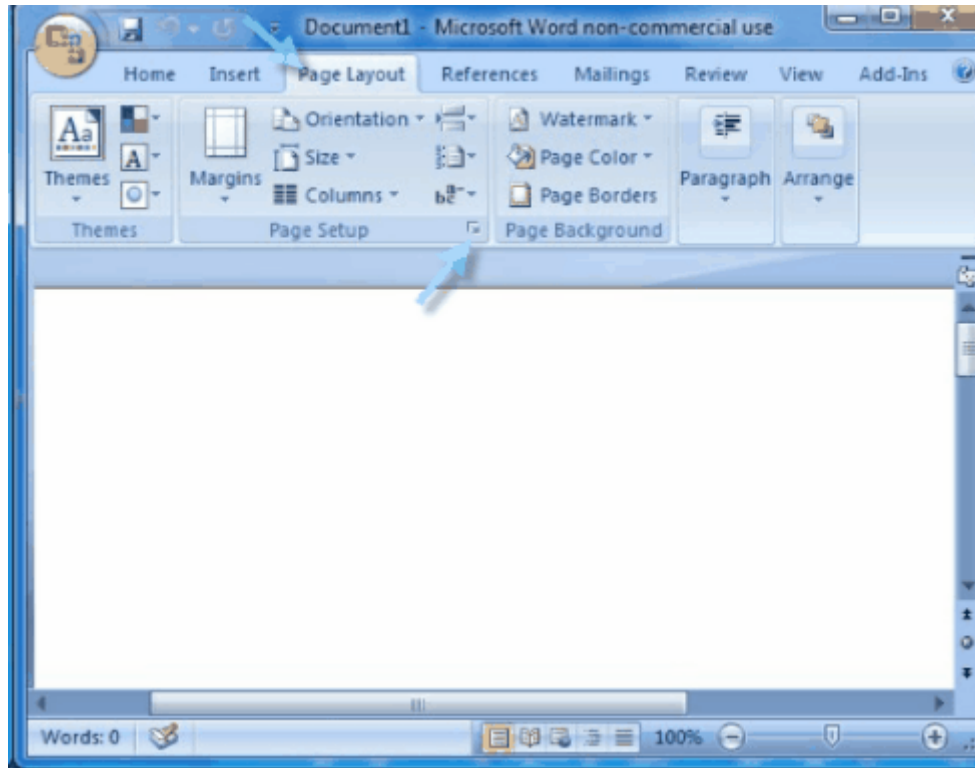
If you have taken advantage of Word's 'Different Odd & Even' and/or 'Different First Page' header/footer settings, make sure that you change all appropriate headers and footers. You may have to add a 'dummy' second page in order to trigger the display of the second page header and footer. You can add this dummy page by pressing Ctrl-Enter.

Sometimes an original document has 'Different Odd & Even' or 'Different First Page' header/footer settings that you don't want or no longer need. You should modify those settings in the Page Layout section of your template. Here are the steps:

1. Click File|Page Setup.
2. Click the Layout tab.
3. Uncheck any undesirable settings.



Getting to the Page Layout screen is a little different in Word 2007/2010, but the same number of steps. Click the Page Layout Tab and then the Page Setup button in the lower right corner. Click again the Layout tab that appears in the resulting menu.



Accessing Page Layout in Word2007/2010.

3.14.3 'StartHere' Location

Regardless of the suggestions previously made that Pathagoras considers template as an essentially 'blank' document which provides 'shape' (styles, headers, footers, etc.) to the clauses that will be poured into it, a template can be much more than that. Indeed, a template can be a complete document which is laden with optional and options text blocks which the user will decide to keep or discard at the time the document is called up for assembly.

Oftentimes, however, a template is something in the middle. It will be a document that provides the shape, etc., and maybe some introductory language.

AND (and this is the purpose behind this page) the template may contain a substantial amount of introductory text. You may want your selected clauses to be inserted not at the very top of the document (the default insertion location) but at a point somewhat below that location (such as immediately after the introductory text).

Pathagoras can oblige. You just need to set a 'StartHere' location in the template. With the 'StartHere' location set in a template, Pathagoras will first lay down the template when you initiate a document assembly session, but before pouring in the selected clauses, it will first jump to the 'StartHere' location.

Pathagoras recognized 3 types of StartHere locations. A 'StartHere' location can be (1) a particular page, (2) a particular section or (3) a specified 'point'. (A 'StartHere point' is marked by a standard Word bookmark which has been named "StartHere".)

You can use the Templates Screen to automatically set the 'StartHere' page, section or point for you. Display the template. and place the cursor at the point you want Pathagoras to begin inserting selected clauses. Then, display the Template Screen and click 'Create StartHere' point

from the available options.



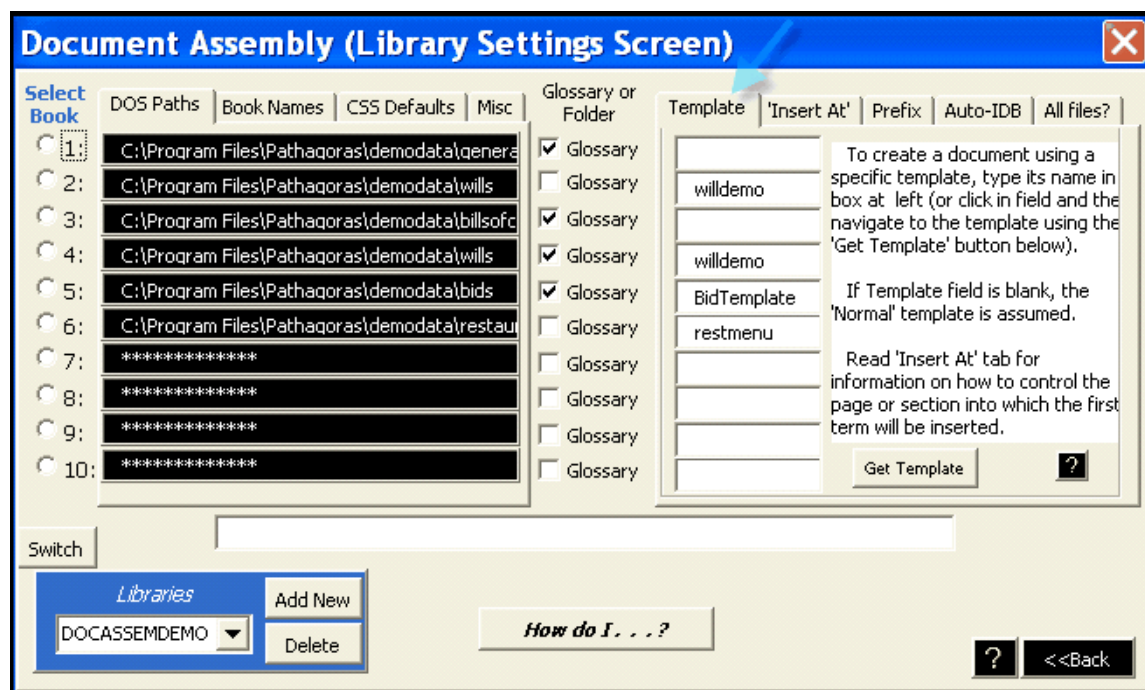
You can also set a StartHere point manually. Recall the template to your editing screen and place the cursor at the point you want Pathagoras to begin inserting selected clauses. Click Insert|Bookmark from Word's menu bar. Name the bookmark "StartHere." Save the template.

3.14.4 Assign Template to Book

Assign a template to a document assembly book.

To assign a template to a document assembly book (this is a one time process only):

- (1) Click the Document Assembly icon (third from the left)
- (2) Click the <Settings> button to display the settings screen.
- (3) Note the 'Templates' tab at the right half of the Settings screen. (See Figure 1.) Type the template name, or navigate to the specific template in the appropriate box. See 'Search Strategy' below for more information.
- (4) Save the changes.



The Document Assembly Settings Screen. Template tab visible.

You can create a new template *and* attach it to one of your books in a single step. Click here for the process.

Search Order for Templates

Pathagoras has a search strategy for finding templates assigned to books. Pathagoras will look for a template in the following order:

- If the template is 'fully qualified,' Pathagoras will look in the designated location and no further. ('Fully qualified' means that the document is pointed to by drive, folder, subfolders(s) and name. E.g., "c:\office forms\estate planning\trusts\trust.dot".)
- If Pathagoras sees only a template name, it will first look in the folder assigned to the book.
- If the template is not there, Pathagoras will look in the default templates folder and in the first level of sub-folders beneath that.

Therefore, you can store the template in one of several locations, and Pathagoras will be able to find it. Do realize, however, that once Pathagoras finds the template, it doesn't look any further, and does not warn you that multiple copies of the template might exist.

Many users prefer to store the template into the folder belonging to the book, because they want to share their work with other Pathagoras users. By storing the template along side of the other source documents, only a single folder needs to be transferred.

3.14.5 Exempting Document from Assignment

Exempting a specific document from use of a 'book-assigned' template: There may be situations when you want to 'exempt' a document from the automatic application of a template assigned to a book. Here are the steps:

1. Call the document it to the editing screen.
2. Click File|Properties from the Word menu.
3. Click the "Custom Properties" tab. Add a new custom property entry called "Template" and assign it the value of "self."



There are a few additional steps to get to "Custom Properties" in Word2007 and 2010. Here they are:

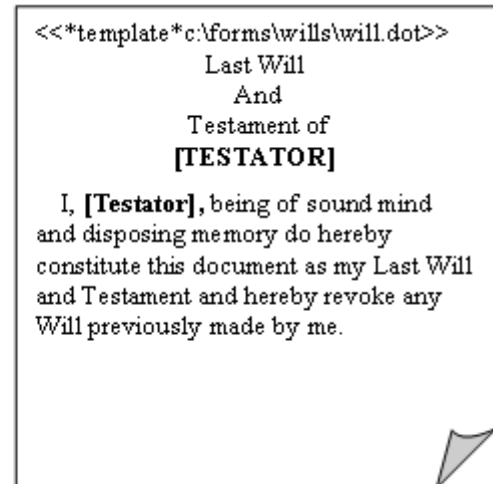
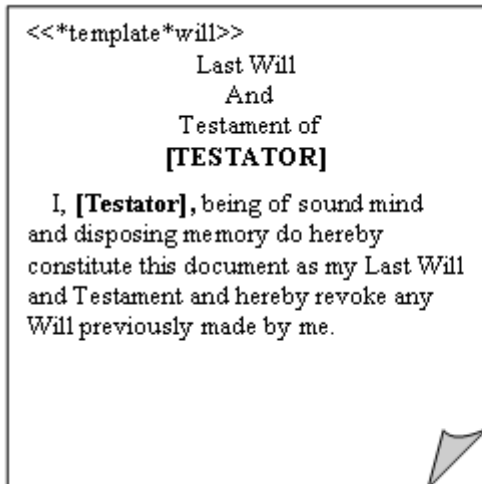
- (1) click the round Office button (upper left of the screen) and click "Prepare;"
- (2) Click "Properties and then click the "Document Properties" drop down list (it sits just above the "Author" entry);
- (3) Click the "Advanced Properties" button and then click "Custom Properties."
- (4) Add a new custom property entry called "Template" and assign it the value of "self."

3.14.6 Assign Template to Document

You can assign a specific template to a specific source document. This allows you to avoid the templates tab in the Settings screen altogether. By doing so you can supersede the assignment made in the Settings screen with regard to a particular book.

To assign a template to a source document, type the characters: "<<*template*(templatename)>>" at the top of the source document. When the source is called during an assembly session, Pathagoras will call in the designated template.

Examples:




Search Order:

In the example at the left, only the 'short' name of the template is provided. Pathagoras will look in the following locations in the following order, and return the first version of the template that it finds:

- If the template is 'fully qualified' (with a drive letter, full path and full name, as in the example at the right), Pathagoras will look *only* in the location specified.
- In the folder which houses the 'book' from which the clauses are drawn.
- In the default User Templates folder (as set in Tools|Options|File Locations)
- In the default Workgroup Templates folder.
- If you have the multi-licensed version of Pathagoras and have assigned a Common Clauses Path ('CPP'), Pathagoras will look in the subfolder of the CPP called 'Templates'.

Because Pathagoras will search for the template in the above fashion, the '(templatename)' can be the short filename that you want to reference. You do not even need to include the extension ".dot" (Pathagoras will assume it and provide it.).

 If the document typically is built by calling in multiple clauses (i.e., 'document assembly' in its truest sense, where you select multiple clauses from the clause selection screen, or are assembling a clause-set), and you prefer this technique, you should place the "<<*template*(templatename)>>" reference at the top of the first clause that is typically called. (Most books have a typical 'first clause' – they are the 'preamble' clauses for Wills and Contracts and the 'heading' or 'letterhead' for letters and other documents.)

3.14.7 Paragraph Styles

Paragraph Styles and Templates

This is Word's rule with regard to *styles* associated with clauses copied from a source and pasted into a receiving document:

The receiving document controls styles.

It is as simple as that. So, if the style names associated a selected paragraph of a source document(s) is identical to a style name found in the list of style maintained by Word for the receiving document, then the style rules of the receiving document will control. Here is the impact of that rule:

(1) Let assume the style in the source document is called 'body text.' It calls for 'single spacing between paragraphs'. Now let's assume that the receiving document also has a style called 'body text' but that style is defined in the styles definitions as 'double spacing'. Double spacing will occur. Why? Because the receiving document always controls.

(2) Let's assume that a style in the source document is called "Special.' It calls for 'single spacing between paragraphs.' Now let's assume that the receiving document has no style called 'Special.' Single spacing will occur, and the style called 'Special' will be added to the receiving documents list of styles definitions.

(3) Let's now assume that a second document is added to the document under assembly and it is added after the document described in (2) above was added. And let's further say that the second document also contains a style called 'Special' but this style was defined as 'double spacing between paragraphs.' When that paragraph is poured into the document under assembly, it will come in as single space. Why? Because when the first paragraph came it, it established the definition for 'Special', and subsequent paragraphs are always controlled by existing definitions.

See Styles in the Frequently Asked Questions for a more detailed explanation of this important topic.

3.15 <<Document Name>> Reference

Whenever Pathagoras encounter text enclosed within "<<" and ">>" (double angle brackets, or 'DABs'), it will attempt to process what it finds in between those brackets. That's why whenever Pathagoras encounters an [<<*Options*>>](#)¹⁴⁶, [<<*Optional*>>](#)¹⁴¹ or [<<*Repeats*>>](#)³⁸² block during a document assembly session, it knows to process the block in accordance with rules associated with each kind of block.

However, Pathagoras will attempt to process *anything* within DABS. When there are "**(Command)**" instruction as above, Pathagoras assumes the special action associated with the word between the "*"s. But when no "**(Command)**" exists, Pathagoras performs other really cool tricks:

- If Pathagoras sees '!' marks on each end of the internal text, it presumes you want to return the value of the '!groupname!' associated with a Repeat call. Pathagoras will return the spelled out value of the number assigned to that !groupname! (See [this link](#)³⁸⁹ for how to return other formats for a Repeat !groupname!.)
- If there are no special characters (asterisks or exclamation marks) Pathagoras will assume that the text within the DAB represent a document call and will locate the designated clause or document following Pathagoras' [Order of Search](#)⁹⁵ rules.

So, if Pathagoras saw <<c:\office forms\real estate\Listing Agreement.docx>> in your

document, Pathagoras will locate the 'Listing Agreement' in the designated folder, and insert a copy of it into the document.

Likewise, if Pathagoras saw <<signature block>> within a document, it will hunt down the document 'signature block.doc (or .docx) in your system, but instead of being limited to a specific folder (as the above example would require), Pathagoras will continue to hunt for the clause, following the 'Order of Search' rules, and if and when found, insert the appropriate text into your document.

Any reference to any text can be placed between angle brackets. It can be a fully qualified document name (as in the first example), an unqualified document name, a glossary term, or a prefix/suffix named clause.



DAB text that references a document can be nested within an <<*Options/Optional*>> block. This opens up a wide range of document assembly possibilities. By properly constructing the Options/Optional block, you can pose a simple question to the end user within the Options block which, when answered, calls in the appropriate document or clause. For example, let's assume the following <<*Options*>> text block resides in the source clause of a letter being written to a purchaser of goods. The purpose of the clause is advise the recipient of the letter what the shipping costs would be in various situations. Assume names within the DABs are existing documents in the user's system.

```
<<*Options(radio)*Order Amount:$0 - $100/$101-$200/$201-$500/$501 + *<<Full
charges>>/<<20pcnt discount>>/<<50 pcnt discount>>/<<Free shipping>>>>
```

When the above text block is encountered during Pathagoras' top-to-bottom 'processing' of the document, Pathagoras will present the 'questions' "\$0 - \$100" "\$101-\$200" "\$201-\$500" and "\$501 +" onto a pop-up form for selection. The user makes the appropriate selection and based on that selection, the appropriate <<document name>> value is returned to the screen (albeit only briefly).

On its next pass through the document, Pathagoras then 'sees' that value and (because it is DAB text), begins to process it. Following the normal [Order of Search](#)⁹⁵ rules, Pathagoras locates and places the desired text onto the screen. It quite literally is blink of an eye fast.



One further addition to your <<*Options/Optional*>> block can add even more power to your source document. You can add a group name. That way, the first answer provided in the top-to-bottom processing of the document will result not only in the insertion of the proper clause at the 'top,' but also for all other clauses in the same !group! in a locations further down in the document. With !groups!, you can create very complex documents, yet pose very simple questions to the end-user.



If you have a DropDown List of clauses that you would like to use to insert <<document name>> references, cycle to the <<Insert Name>> option in the upper left quadrant of the DropDown Lists section.

iNote: The "<<" and ">>" marks are not special keyboard or ASCII code. They are the 'lesser than' and 'greater than' characters above the 'comma' and the 'period' on your keyboard.

i While not precisely the same, implementation of above can bring powerful "If . . / Then . . ." logic to your documents. See a further discussion of this at [this link](#)⁴⁹².

See Also:

[Suppress Processing](#) 158

[Order of Search Rules](#) 95

[Clause Sets](#) 406

[If... / Then... logic](#) 492

[Instant Database: Double Angle Bracket Functions](#) 244

3.16 Tree Service

Take advantage of all documents within a folder tree!

Tree Service allows you to select clauses not only from the parent folder, but from any folder 'below' the parent. Tree Service allows you to have a larger variety of clauses and documents immediately available to you during the clause selection process. and.

Turn 'Tree Service' ON by clicking the 'Tree' checkbox on the Clause Selection Screen (located just beneath the left panel). The Clause Selection Screen will repaint, showing additional clauses (or paths to clauses) as discussed in the next section below:

(Note: this is a relatively advanced feature of Pathagoras. You should be familiar with basic document assembly techniques before implementing Tree Service.)

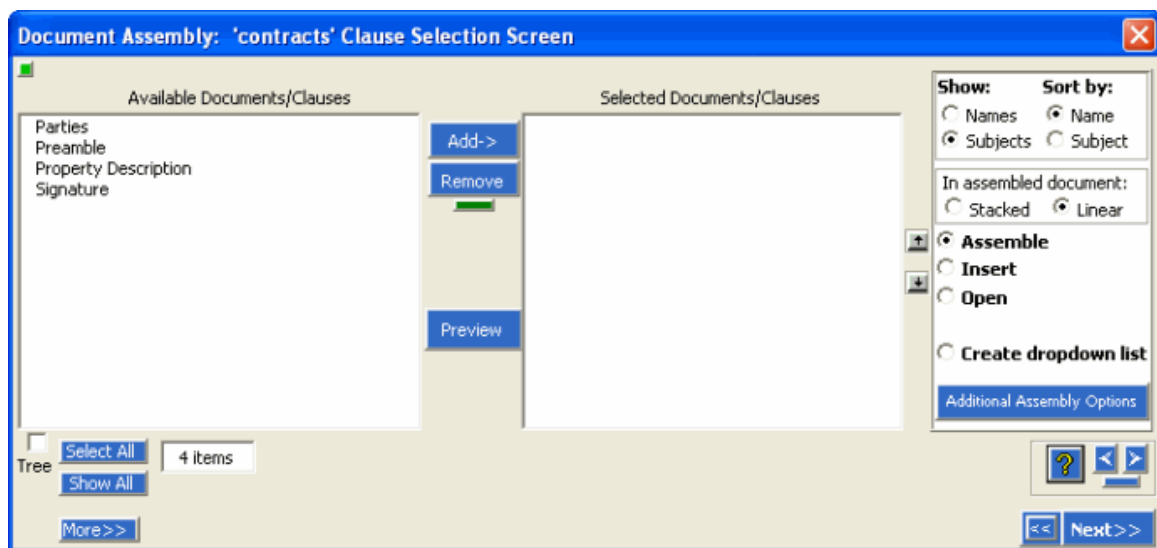


Figure 1. 'Standard' Clause Selection Screen ('Tree Service' not selected)

The above figure shows the standard view of the Clause Selection Screen. The folder that is open shows a few clauses that might be selected for a real estate contract.

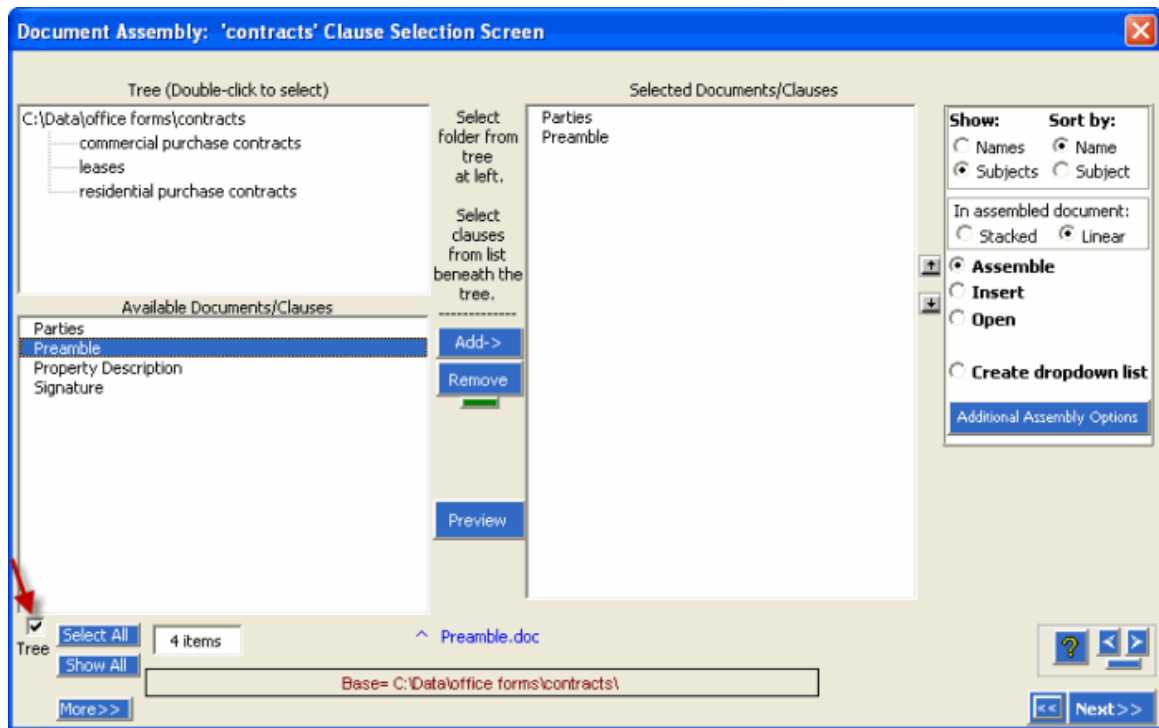


Figure 2. 'Tree Service' activated.

In this view, a few clauses were selected, but more are needed. The Tree Service has been activated by checking the box labeled "Tree." The screen is automatically repainted to include the sub-folders beneath the parent 'book.' The tree appears at the top-left of the screen.

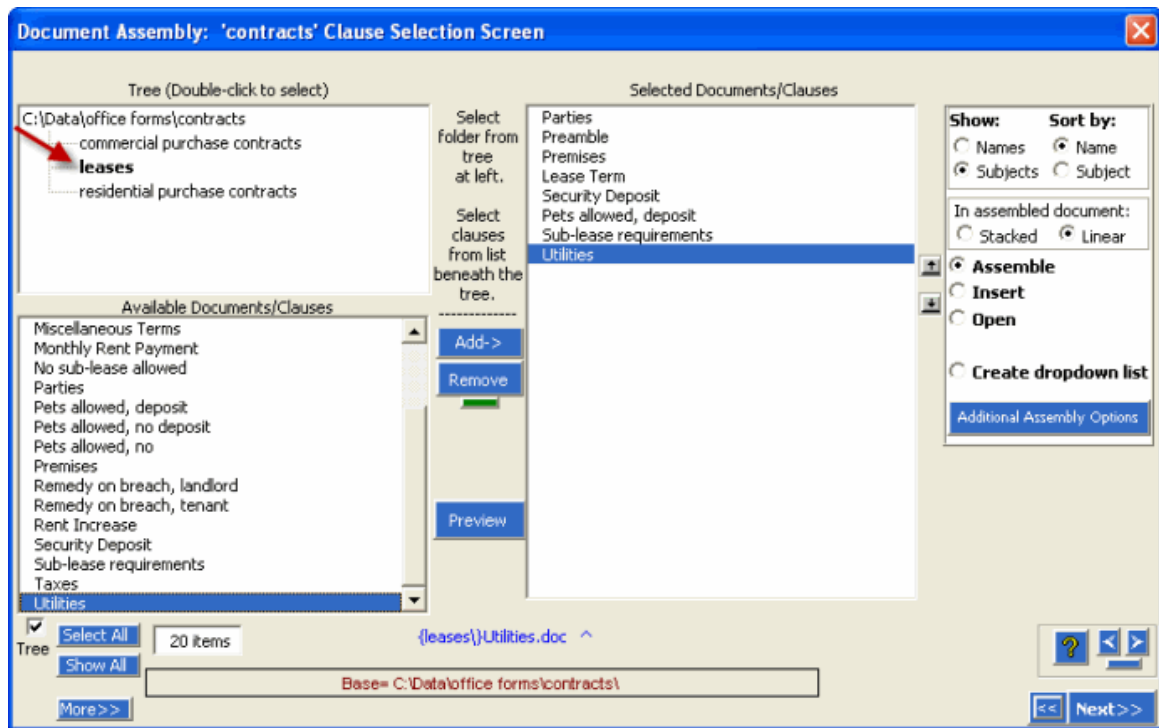


Figure 3. Select desired clauses (if any) from the sub-folder.

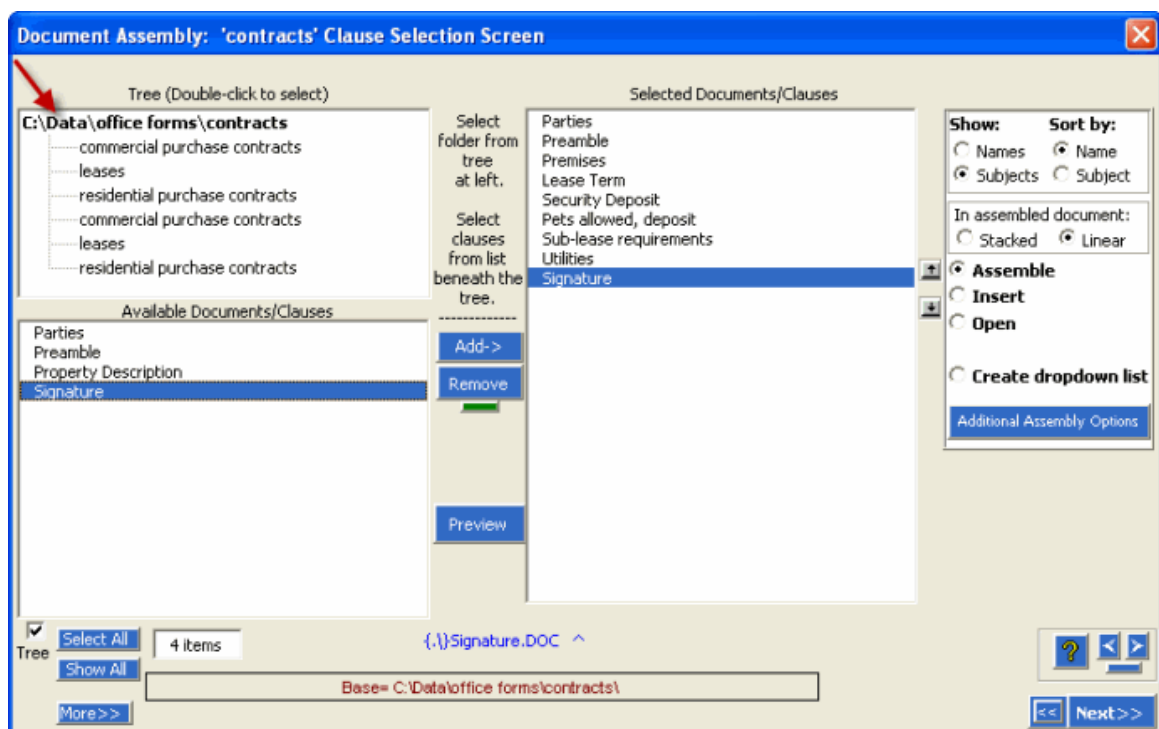


Figure 4. Navigate to any other folder (including the parent folder) and choose content.

Make selections to complete final document.

Select clauses from any (or every) sub-folder as appropriate. Note the {.\} and {\Leases\}

prefixes at the right. The text in the braces indicates from where in the tree the clauses were pulled. {\.} means the 'root' (base) level. The other entries denote the sub-folder(s) from where the clause came.

Glossaries vs. Folders of Clauses

Tree Service works slightly differently depending upon whether clauses are saved in a glossary (all clauses in a single document) or as separate documents in a folder.

If the book is a folder of documents:

If there are subfolders beneath the current book, the name(s) of the subfolder(s) will appear in the tree area.

If the book is a glossary:

Pathagoras 'tree service' will display all glossaries that are in the *same* folder as the primary glossary (the one assigned as the book.) When you activate 'Tree Service,' the name(s) of any glossaries in the same folder will appear between braces and in all {CAPS} in the left panel. If you double click on a {GLOSSARY}, Pathagoras will repaint the Clause Selection screen and you will see the files in the selected glossary, plus the other glossaries (including the parent glossary) in the same folder.

Common to both:

When you double click a file or clause (or highlight the item(s) and press the Add>> button), the selection will be moved from the left panel to the right. A prefix within braces designating the source of the clause will be appended to the clauses name. E.g., "{subfolder}Name of Term" This may at first be a bit disconcerting to the user, but it is a necessary aspect of the routine in order to show the source of the clause. (If you have selected 'Subjects' as your primary display, you will not see the {prefix} because it is off screen in the right panel. You can, however, see it at the foot of, and centered between, the two clause selection panels in blue text.)

• DropDown Lists

The tree service works with Pathagoras' DropDown Lists in the same way that it works with the Clause Selection screen (but not yet with glossaries, only folders). To activate the tree service, you must recreate the Drop Down list. You will see {SUBFOLDERS} at the top of the list. Click on a subfolder and the list will repaint with the files in the subfolder.

• Set Tree Service as Default Display

You can set Tree Service to be 'on' by default. The setting is in the Document Assembly Settings screen. (Click Document Assembly icon and then click <Settings>.) Click the "CSS Defaults" (Clause Selection Screen) tab and look for the 'Tree View as Default' checkbox at the bottom of the list.

• Clause Sets and Tree Service

Clause-sets can easily be created using Tree Service features. Indeed, you can create incredibly complex documents in a matter of seconds when you can pull from a large variety of clauses in a variety of folders.

• Does it matter which Glossary is the Base glossary?

Since Tree Service gives me access to all glossaries in the same folder, does it matter

which one I activate? Maybe. The reason why you may prefer one glossary over another is to insure that the template or IDB mask that was associated with the particular glossary is activated. If there are no template or IDB issues, then it doesn't matter which is the primary glossary.

3.17 Tables and Pathagoras

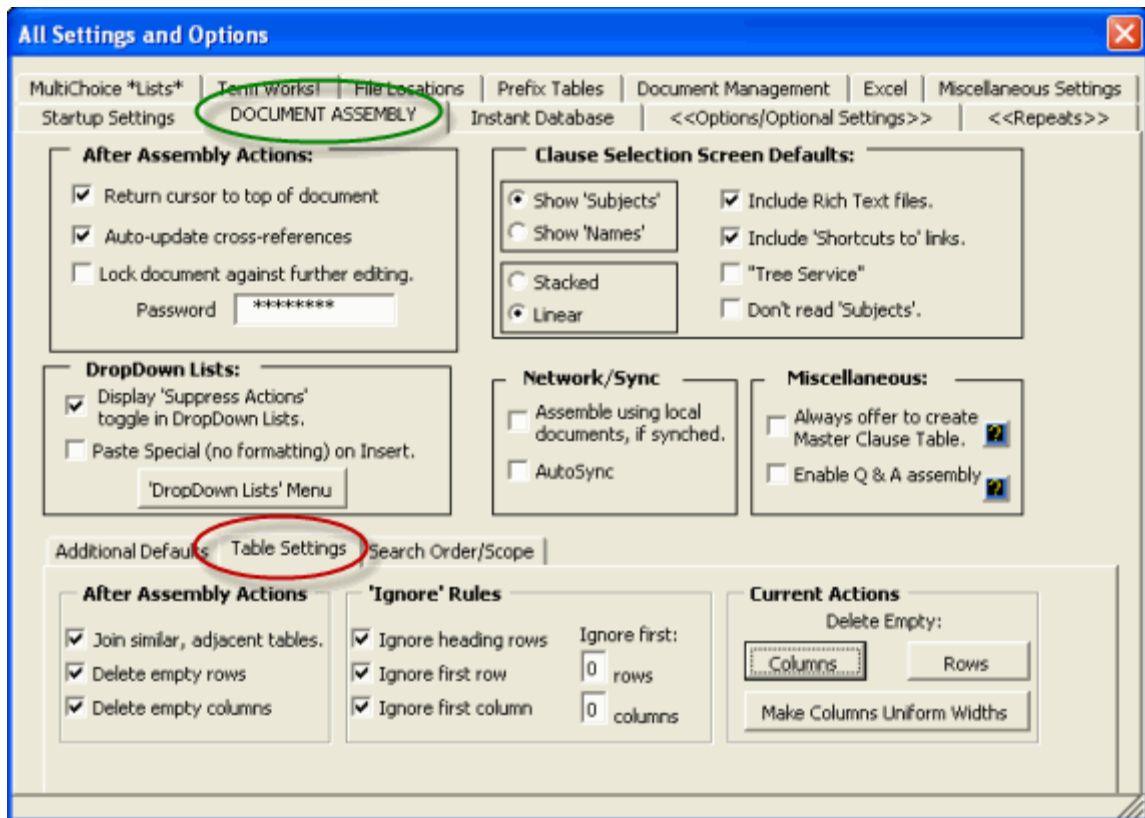
Pathagoras and tables work well with one another. For example:

- you can insert variables into any cell of any table.
- you can insert <<*Options/Optional*>> text blocks into any cell of any tables.
- you can assemble tables stored in source documents.
- if a series of otherwise adjacent tables (separated by a single 'enter' after assembly) appear in the assembled document, you can join those tables into a single table.
- if any table has empty rows or columns, Pathagoras can remove those rows or columns, and even resize the individual cells to restore the original tables original width.
- you can repeat tables 'X' number of times.
- you can repeat rows within tables 'X' number of times.

Here are some other tools involving tables.

Join Similar Tables:

If following a document assembly session you have a series of 'almost adjacent' tables (and by this, we simply mean tables that are separated by a single paragraph marker), Pathagoras can pull those tables together into one.



The Join Tables and ancillary command can be found under the Document Assembly tab of the "All Settings and Options" screen. (Click Table Settings in the lower tab set.)

Limitations:

- To be joined, Pathagoras requires that the last row of the upper table be identical in shape to the first row of the lower table. If you want all tables to be identical, use the Make Columns Uniform tool.
- Tables separated by more than a single 'enter' mark cannot be joined. (You can turn this into a 'plus' if you want to set 'Join Adjacent Tables' as your default setting but want to keep certain tables that would otherwise qualify for joining. Simply type a 'space' in the source document before the closing 'enter' mark. This will prevent the joining because more than a single 'enter' mark would separate the two tables.)

Delete Empty Rows & Columns:

As a result of making [Variable], <<*Options/Optional*>> or {Optional} text block selections, you may end up with a totally empty row or totally empty column. You can pre-designate how you want Pathagoras to handle these empty rows or tables. Check the appropriate After Assembly Actions that you desire. Caveat: Use with caution. As discussed below, this may result in some unexpected behaviors, especially if your tables are not consistently formed across your entire library of forms.

When is an empty column or row really 'empty'? Frequently the top-most row or left-most column contains header or titular information. If the row or column is otherwise blank, you want the entire row or column to be deleted. In such case, check the 'Ignore First Row' box. When so set, Pathagoras will not check the first row to see if the top cell is blank. (If you need to ignore more than the first row, a rarity, but possible, you can set the count of how many rows to

ignore to whatever you desire.) Pathagoras will only examine the remaining cells in that row or column to determine its 'blankness'. But be careful with this setting. Pathagoras cannot distinguish among tables. If you have one table for which you wish to ignore the 1st row (or column) and another for which you don't, Pathagoras will not know that. It will handle all tables in the same fashion. But see Note: below as to a way 'around' the rule.

Note: there may be situations where you want to preserve certain empty columns or rows even though you have set "Delete Empty Column/Row" as your default. The 'fix' is easy. Just type a 'space' in the blank cell and it is no longer 'empty.' If the cell is made 'blank' as a result of a [variable] or <<*Options/Optional*>> text block selection, set the variable or the result of the option selection a ' ' (space) if a 'no text' option is chosen.

Note 2: Turn off the Delete Empty Column/Rows feature if your tables are not consistently formed (all with title rows and/or columns, or all without). This feature, as 'cool' as it is in many settings, will cause more headaches than it will save time when your tables are not consistently formed.

Note 3: Pathagoras can recognize a 'true' heading row from the first row which coincidentally contains heading information. Generally a 'true' heading row is shadowed or is of a different font. If some of your tables have true headings and others do not, and you want to implement the 'Delete Blank Columns' feature, you can check the Ignore Heading Rows and not run across the problems cited in the above discussion.

Repeat Tables and Repeating Rows within Tables:

These topics are discussed more fully in the section titled [Repeats and Tables](#)³⁸⁷.

Making tables uniform in shape.

Note: When the Make Uniform routine is invoked, Pathagoras will ignore the first *X* rows of the first table only. Pathagoras presumes that the first *X* rows of the first row of the document (if whole document selected), from the cursor point downward (if selected) or the highlighted text (if selected) reflect the header, and, further, that the header does not repeat in subsequent tables that you intent to join. To the extent that this does not reflect the reality of your situation, you should copy the tables that you want Pathagoras to operate on into a separate document, perform the table adjustments and then copy the tables back into the original document.

Ad hoc removal of empty rows and columns:

The above lets you set Pathagoras to automatically join similar rows or to automatically delete empty rows and columns. If you don't wish these as 'automatic' functions, you can still accomplish the tasks more manually.

- (1) From the screen shown above, click one (or both) of the buttons at the far right to delete empty rows and empty columns, as desired or
- (2) From the Pathagoras Features dropdown list, look under 'Process' tools. Click the appropriate button(s).

3.18 'Tag Along' Clauses

In the 'typical' document assembly session, the source text for augmenting the document currently under 'construction' is stored in books (folders or glossaries) that are external to the

current document. They are saved in this manner because the system administrator expects that the clauses will be used over a fairly broad scope of documents or will be used with such a high level of repeat calls that storing them in a separate folder (thereby making them available as a book and/or a DropDown List to others) makes sense.

Sometimes, however, certain clauses are going to be used with a particular document and with none others. For these few clauses to be stored in a separate folder and assigning an entire book to them (or assigning them to a DropDown List) seems an inefficient use resources. In such cases, it just is easier and more efficient to include these optional clauses within the source document itself.

We call such text 'tag along clauses.'

These clauses 'tag along' with the document and when needed, can be quickly inserted at the appropriate location in the document. The entire document (including the optional 'tag-along' clauses) is self-contained. It can be sent to another user without having to send the folder that (otherwise) would contain the 'external' text.

While clauses can still be drawn from external sources, a document such as is described above can be made entirely self-contained, without any need to call a clause from any external source.

Tag along clauses would be most frequently used when the clauses will never be used outside of the particular document. In such cases, the user or administrator may not want to devote a book or DropDown List to housing these clauses. It would also be appropriate to use 'tag along' clauses when you want to send a document to another Pathagoras user who is not on your network. That way, the other user won't have to create a new book or DropDown List to access the additional (optional) text. It is already available in the transferred document.

There are two requirements for Tag Along text to function in the stand alone method.

1. Each optional block of text must be bookmarked. That is the only way to provide quick access to the individual clauses. To bookmark the block of text:
 - To use Pathagoras 'highlight & add' routine, highlight the text you want to bookmark with your mouse (or use keyboard controls) and then press Alt-G. Provide a name for the term in the TermWorks screen and then click 'Other Book' and check the 'TagAlong' box.
 - To manually bookmark a section of text, highlight it with your mouse (or use keyboard controls). Then click the Insert button in your Word toolbar area and then click Bookmark. Provide a meaningful name for the bookmarked term. (Bookmark naming rules require that the name start with a letter. Only single words are allowed, but you can use the underscore character to 'separate' words. Press the Ok button to save the bookmark. Repeat with each separate clause.
2. You need to tell Pathagoras where you have 'stored' your tag along terms. Do so by typing **&Tag Along Clauses Start&** and **&Tag Along Clauses End&** to each end of the section of your document that contains your tag along terms. (We recommend placing your tag along terms at the bottom of your document.)

If Tag Along terms reside in the document being processed, Pathagoras will locate them (using the boundary tags that you inserted in step 2 above) and present the individual clauses in a small overlay screen. The end user can select one or more (or none) of the clauses. The text will be inserted where the cursor was last clicked in the document.

At the end of the document assembly process, the user will probably want to delete the tag along clauses. (They probably should not remain in the final document sent to a 3rd party.) Deleting the tag along clauses is the last choice in the list.

The Pathagoras System

'Pathagorizing' Your Documents (Variables and Optional Text)

Part

IV

4 'Pathagorizing' Your Documents (Variables and Optional Text)

We call the process by which a document or form is prepared for use by our program: 'Pathagorizing'. Actually, it is really called 'neutering,' but we had so many complaints from dog owners that . . . well, you get the picture. And it is also called 'sanitizing.'

Regardless of what you call it, the concept refers to removing actual names, addresses and other personal information from a complete document and adding [variables](#)¹¹⁷ and [<<*Optional/Options*>>](#)¹⁴¹ text blocks. These elements turn a 'plain' document into an 'automated' document. .

'Pathagorizing' is a phrase coined by us. Do not look for it (yet) in a dictionary.

The goal of 'Pathagorizing' is to create documents that:

1. contain no text that could identify an existing specific client, patient or customer and
2. can be quickly and accurately and professionally personalized.

The more neutral the source document, the less the chance that there will be a noun, pronoun or verb that would be inappropriate in your new product, or that personal information that could violate a client or patient privilege would reside in the initial drafts.



Pathagoras provides many tools to automate adding variables and [<<*Optional/Options*>>](#) text blocks to your documents. But, as you are Pathagorizing text, keep in mind that you are always working with a standard Word document. The positive consequence of that can be summarized this way:

- When you are creating variables, you can ignore all of the automation tools. You can Pathagorize any document just do so by typing the appropriate boundary text from your keyboard. [Customer Name] is a variable whether you type in free hand, or insert it into your document using another technique that Pathagoras provides.
- If you have a variable that you want to place in multiple spots throughout your document, type it once. Highlight it. Copy (Ctrl-C will do it) it into your clipboard and then paste (Ctrl-V) it where you want it. (Yes, Pathagoras can do this for you through its various tools, but you should take comfort in knowing that there is nothing 'magical' about what it is doing.)
- If you are Pathagorizing an existing document with a name (e.g. "John Doe") throughout the document, use the Find & Replace tool to find John Doe and replace it with [Customer Name]. Same result.



Editing: Because all text needed to create any variable is 'plain text', you do not have to have Pathagoras loaded on your system in order to create them. You can create and fully edit them anywhere. This allows 'at home' and third party editing.

See Also:

[Variables](#)¹¹⁷

['Create Variables' Assistant](#)  122['Create Options' Assistant](#)  155[Drag and Drop \(Using existing Instant Database masks to insert variables\)](#)  131[Optional Text Blocks](#)  141[Editing and Debugging: Balancing Markers](#)  446[Document Disassembly](#)  330[One Page Summary of Markups](#)  176[How does a 'variable' differ from 'Optional text'?](#)  628[Paragraph Assembly \(clause snippets\) vs. Assembly using Templates](#)  622

4.1 Variables - Definition

A 'variable' is a place holder for personal information. Variables are placed in strategic locations throughout a document to indicate where personal information ultimately will be placed in the final version of the document.

A '*simple variable*' is a single word or short phrase that serves as a place holder for something else. Typically that 'something else' is personal data such as a name or an address, but it can be much more complex than that. In Pathagoras, a simple variable is a word or phrase all enclosed between square brackets. E.g.: [ClientName]

A '*multiple choice variable*' is a series of pre-determined choices from which the end-user may choose. In Pathagoras, the various values of a multiple choice variable are separated by slashes. E.g.: [chocolate/vanilla/strawberry].

The following pages describe in greater detail the process of creating, naming and using variables.

4.2 Creating Simple Variables

Creating variables with Pathagoras could not be easier. You use only standard keyboard characters.

If you are starting from scratch, type an opening square bracket, a variable sounding name and then a closing square bracket.

If you are Pathagorizing an existing document, peruse the document for places where personal information current exist. If the text is currently a personal name, change it to a variable sounding word and then put brackets around it. So, change "Abraham Lincoln" to "[FirstName] [LastName]" and you have done it.

These are variables: [Client Name] [Quantity Ordered] [Date of Contract].

Note the following:

- You can manually add variables by hand typing them into your source documents.
- If you work with documents that you obtained from a third-party which already contain bracketed variables, you actually have a 'Pathagorized' document. (And you can add

even more variables to those documents if you wish.)

- If the same name (e.g., "John Doe" is in numerous places throughout the document, you can use Pathagoras' [Create Variables Wizard](#)¹²² to convert each instance of personal text into the desired variable.
- When deciding on variable names, keep the end user's reaction to the variable in mind: "[Date Contract Signed]" is probably better than "[Date]."

Check out this video on creating and using simple variables:

[Creating Simple Variables](#)

Modifying variable names: Sometimes the name you initially select for a variable turns out not to be the 'best' choice. Not to worry. You can easily change them to a more appropriate value. If you are just getting started, and have only created a few variables, manually changing them is probably the easiest course. But if you have spent a lot of time creating variables before you decided a different name(s) was in order, see [Search & Replace](#)⁴⁴⁸ for suggestions on how to make those changes in a global fashion.

See Also [Converting HotDocs variables](#)⁶⁰⁸.

4.3 Creating Multiple Choice Variables

A multiple choice variable is a simple variable containing a series of words or phrases that you want to offer to the end user as separate choices. Each choice is separated from the others by a plain text slash ("/").

Examples of Multiple Choice Variables:

[he/she/it/they] [his/hers/its/their]

[red/white/blue]

[chocolate/rocky road/vanilla cream swirl/triple fudge brownie delight]

[Poughkeepsie/Patchogue/Irondequoit/North Tonawanda/Skaneateles]



A multiple choice variable is typically used when the author wants to provide the end user with a set number of possible 'answers.' But, as the last example above illustrates, spelling challenges can be another reason to provide a multiple choice variable.

Usage examples:

When the entire range of colors is a possible answer, a simple variable like "[color of widgets]" would work fine.

However, if you wish to limit the selection of colors to just a few choices, you can list those colors within the variable itself as a multiple choice variable. That way, the end user understands the limitation, and will be led to provide an acceptable answer:

Thank you for your order. We will deliver [number] dozen of [yellow/green/blue/orange] widgets within 2 weeks.

When Pathagoras encounters a multiple choice variable, it parses out the individual choices and presents them in an easy to select drop down list (in the [Instant Database](#)¹⁸⁶)

module) or selectable buttons (in [GotForms?](#)³⁴⁸). Try it out. Type a simple multiple choice list onto your editing screen. (You can even copy and paste the examples above.) Press <Alt-D> to activate the Instant Database. Press the <Scan> button. The variables appear at the left, the dropdown lists at the right.

What if I have a 'lot' of choices?

Pathagoras allows you to create an unlimited number of choices within a standard multiple-choice variable. However, if you need to list more than 5 or 6 we recommend that you use [Aliases and *Lists*](#)⁴⁵⁸, a complementary feature discussed in the next section that allows you an unlimited number of choices. (Think the 50 United States, all the countries in the world, all of the flavors of Ben & Jerry's® Ice Cream, and similar long lists.)

What if I want to use a multiple-choice variable in different sections of my document, but be able to select different values?

Making unique names for each variable is easy to do for a simple variable. Just use a different name, or at the very least, append a number or other character at the end. E.g., "[Variable1]", "[Variable2]", "[Variable3]" etc.

To make otherwise identical multiple choice variables 'unique', append the '@' sign and a distinguishing *single* character or *single digit* number at the end of the list of choices. E.g., "[Red/Blue/Green@A]", "[Red/Blue/Green@B]", "[Red/Blue/Green@C]", etc. (Pathagoras knows not to include the distinguishing character or number as part of the last element of the multiple choices when it encounters the '@' sign.)

What if a slash is an integral part of the variable?

If need to choose among variables that themselves contain slashes, e.g., "1/2" and "and/or" put the whole choice in quotes to tell Pathagoras not to process them. For example:
["1/4"/"1/2"/"3/4"/1 inch]

Can I include a picture or a Word 'field' (those 'grey' embedded things) as one of my choices?

No. Variables are 'plain text' in nature, and therefore Pathagoras can handle only choices that are typed as plain text and which can be stored as a permanent record in plain text. HOWEVER (and probably better for most purposes) you can present pictures, Word fields and a wide range of other non-text objects within an <<*Options*>> or <<*Optional*>> text block. This powerful alternative should accomplish what you are seeking. See [<<*Options*/*Optional*>> text](#)¹⁴¹.

What if I need to repeat the same variable selection in a different section of my document, or make a choice in another section that is based on my initial selection?

Not a problem. That is what [!Groups!](#)¹¹⁹ (next section) is all about.

See Also: [MultiChoice *Lists*](#)⁴⁵⁸

4.3.1 Groups

Sometimes, the value assigned to one multiple choice variable suggests the answer to a subsequent multiple choice variable. For example, if "he" is selected as the value for multiple choice variable [he/she/it]. then, "he" (or "him" or "his") likely will be the selection for another multiple choice variable further down in the document if the same

'actor' is being referenced.

For example:

[He/she] went with [his/her] dog to the pet store because [he/she] wanted to buy [him/her] a new collar. Blue is [his/her] favorite color.

(Okay, grammarians, the last [him/her] and [his/her] sets could be better structured to prevent noun confusion, but just play along with me here.)

Note that the first decision quite directly suggests the responses to the next two. (It does not suggest the response to the third and it may or may not suggest the response to the fourth).

Without a way to group the variables, the typical action of the Instant Database replacement scheme would assign the selection of the [he/she] multiple choice to all appearances of [he/she] throughout the document. That result, of course, is not 'okay.'

Pathagoras provides the 'way.' Simply add a short prefix to each related variable. We call this prefix a 'group name' and this group name allows you to connect related sets of multiple choice variables.

To add a group name to a multiple choice variable, type a letter, word or phrase between exclamation marks at the very beginning of each variable in the document you want to be in the group.

e.g., [!groupname!variablename].

So the above could be rewritten:

[!Owner!He/she] went with [!owner!his/her] dog to the pet store because [!owner!he/she] wanted to buy [!dog!him/her] a new collar. Blue is [!owner!his/her] favorite color.

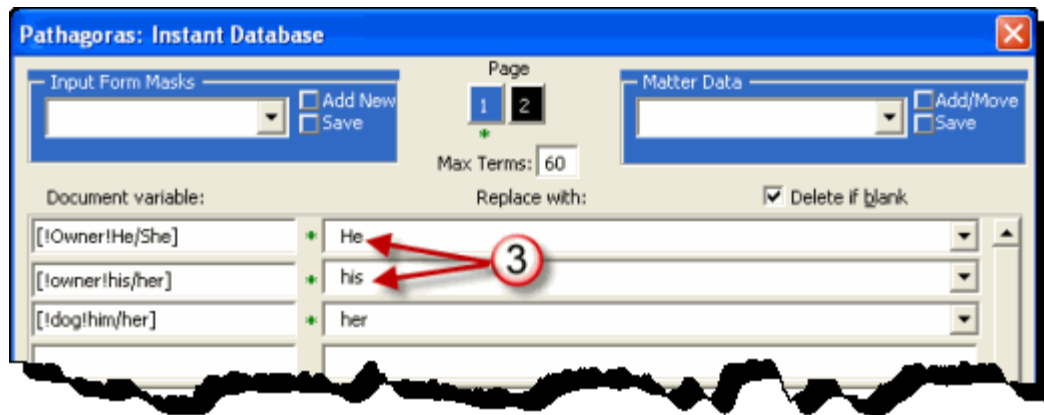
(Perhaps the last group name is supposed to be "[!dog!his/her]"?)

Once assigned, a '!group!' links together all variables within the same group. The result is that the selection made for the first member of the group will trigger the answer to the remaining members of the group. This is so even when the variables are not identical. (The above example illustrates that. Of course, "his/her" is not identical to "he/she".



After the document is scanned, the multiple choice provided appear in a drop down list at the right. Drop down the list (1) and make a selection (2).

The other members of the group (but *only* of the particular group) are automatically chosen (3) below.




As you are designing groupings, don't limit yourself to simple gender choices such as those above examples. Think big!

The [!actor!man/woman] went to the [!actor!ladies'/men's] section of the department store to buy a [!actor!skirt/pair of pants] for [!actor!his/her] [!actor!wife/husband].

Try it. Copy and paste any of the above examples into a document. Run Instant Database (<Alt-D>) against them and see the results.

Keep the following in mind:

 Keep the following concepts in mind:

- Groups are not limited to just pronouns. As the above examples illustrate, you can use group names for any multiple-choice collection.
- While an answer to the one element of the group needs to be provided by the end user, it does not matter which group member you click first.
- The result displayed for group members is typically based on the *position* of the answer in the list. It is not a calculated value.
- The group name can be anything. Short is better. A single word, or even a single letter, will work.
- The group name being what Word sees to determining the 'case' (ALL CAPS, Upper And Lower, lower case) of the replacement text, of the group name will control. The examples provided above illustrate this concept.
- Groups are not limited to just variables. You can use the !group names! method for <<*Options*>> and <<*Optional*>> text blocks. [Click this link](#)¹⁵¹ for more information.

4.3.2 Titles

If you have a series of multiple choice variables that are essentially identical, but which you don't want replaced 'across the board' as if one answer applied to all, you can 'title' the individual variables by adding a one or two word identifier at the 'front' of the variable, followed by a colon.

E.g., [Title:choice1/choice2/choice3]


Each separately titled variable will appear a 'stand-alone' to Pathagoras. The title will

appear not only at the left of the IDB screen but at the top of the choices list as well.

Illustration:

[Special Order:Yes/No/Not Applicable]

[Priority Mail:Yes/No/Not Applicable]

 **Note:** Every variable potentially could be 'titled,' but we suggest that you not title a variable unless it's really necessary. A title takes up space in your document and adds little to the understandability of the variable beyond the narrow situation described above.

4.3.3 Sharing Groups

!Groups! can be interchanged between Options and the Instant Database.

For example, if you <<*Options*!state!*States*>> and elsewhere in the document you had [!state!*States*] or even [!state!*Capitals*]

What is being shared is the position, not the value. So if Indiana were selected as the <<*Options*!state!*States*>> choice, "Indianapolis" would be returned for the variable [!state!*Capitals*]

4.4 'Create Variables' Wizard

Using the Create Variables Wizard.

When the document you want to Pathagorize is filled with 'personal data' (names, addresses, sizes, colors, etc.), call on the 'Create Variables Assistant. It allows you to quickly convert each piece of personal data into a [variable].

By way of example, let's assume that you wish to sanitize a Will that was originally written for, and has lots of instances of "Janet Q. Doe" and her husband "James R. Doe", her children, "Meenie, Minie, and Mo Doe", etc. Because what you had created is a perfect example of a well constructed Will, you wish to turn it into a 'source document' for future Wills.

The Create Variables Wizard is one of those 'must know' features about Pathagoras. It turns what otherwise might be an arduous process of 'sanitizing' your personal text-laden documents into into a cake walk. (Here, we use 'sanitize' to mean "clean out all instances of personal text, replacing them with variables.")

To activate the 'Create Variables' Wizard, press the hot key combination <Alt-V> (for 'Variable'). You will see this overlay, planted in the upper right corner of your editing screen (hopefully out of the way):

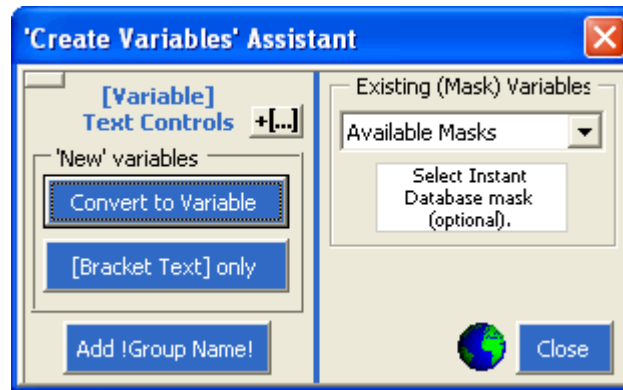



Figure 1. The Initial 'Create Variable' Assistant Screen.

Note that there are two sides to the Create Variables Wizard screen, 'left' and 'right', divided by a blue bar. We will discuss these separately in the following pages.

 As an alternative to Alt-V, you can activate the Create Variables Wizard, by selecting 'Authoring/Editing Tools' | Wizards and Assistants" section in the Pathagoras drop down features list. (Image 2003, and before) (Image 2007 and after)

See Also:

[Variables](#)  117

[Masks](#)  260

['Create Options' Wizard](#)  155

[IDB Drop Down Lists](#)  129

[HotDocs conversions](#)  608

[Drag and Drop](#)  131

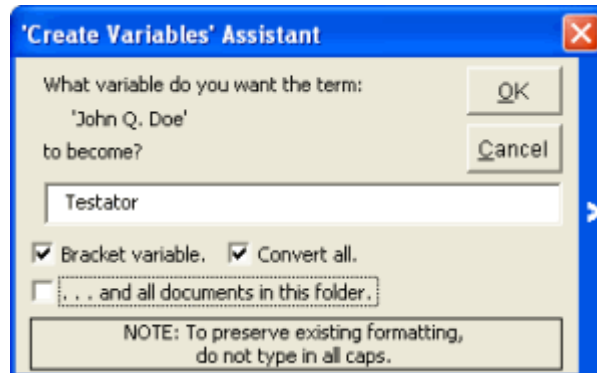
4.4.1 CVA Creating New Variables

The left side of the Create Variable Assistant is used to clean up a document that is currently 'personal' in nature. Its use is best suited for those situations where you have no existing variables in a mask (variables list) from which you wish to draw. That is, you intend to create variables 'from scratch.'

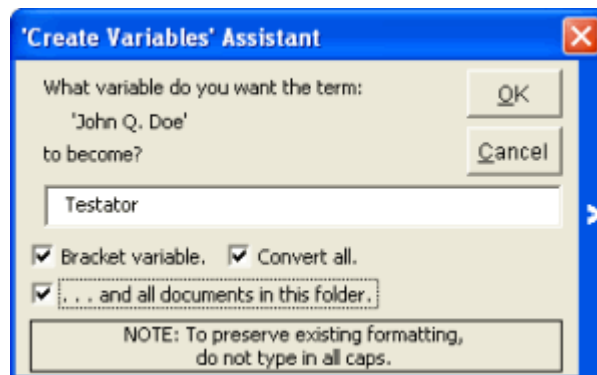
Here are the steps:

1. Highlight the first piece of personal information in the document (a name, an address, color, date, etc.)
2. If the Create Variables Assistant is not yet displayed, press <Alt-V>.
3. Click "Convert to variable" button.
4. Pathagoras will ask you to provide a variable name to replace the personal

information. Type something like "Name of Customer" or "Address of Client" or "Shipping Date" or whatever is appropriate. (Don't include the bracket. Pathagoras adds that for you automatically.)



5. Pathagoras will replace the highlighted text with variable. It will automatically be enclosed with brackets (assuming you did not uncheck the option to do so). If you directed Pathagoras to replace all other instances of the 'highlighted text,' the program will do that for you as well.
6. If you want to tell Pathagoras to replace variables not just in the opened document, but in all documents in the folder containing the document you are currently editing, check the '... and all documents in this folder' box.



7. Continue replacing personal data with variables. The CVA will remain on screen. If it gets 'in the way' you can easily drag it to a corner of your screen.'

i If the document you are working on already contains variable-type names (as opposed to personal data), you can press the '[Bracket Text] only' button.



If the personal information is a single word, you need only place your cursor within the word. Pathagoras will expand to include the whole word. Further, if, as you highlighting text, you picked up the space at the end of the last word, don't worry about backing it out. Pathagoras will automatically back it out for you.

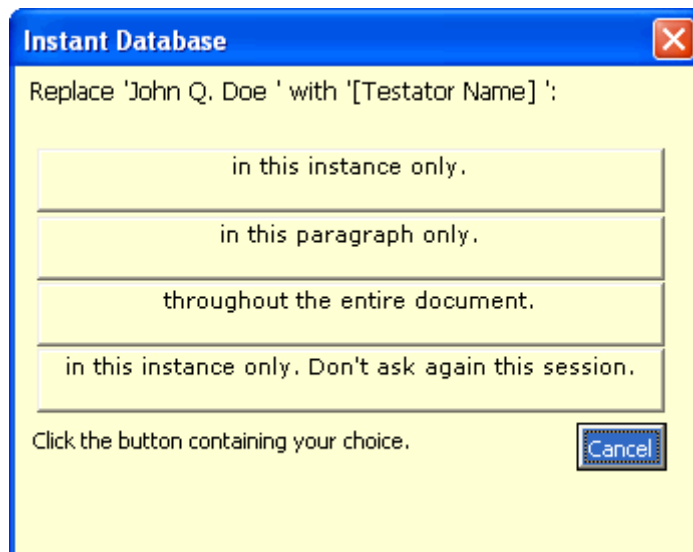
4.4.2 CVA Using Existing Variables

The right side of Create Variables Assistant is for those situations where you want to add one or more 'existing' variables. These are variables that you previously saved into a

mask. (A 'mask' is an collection of variables, typically along with instructions that you -- or we, if you are using one of our pre-loaded masks -- may have provided to guide the end user as to how to complete each variable. (See [Masks](#) ²⁶⁰.)

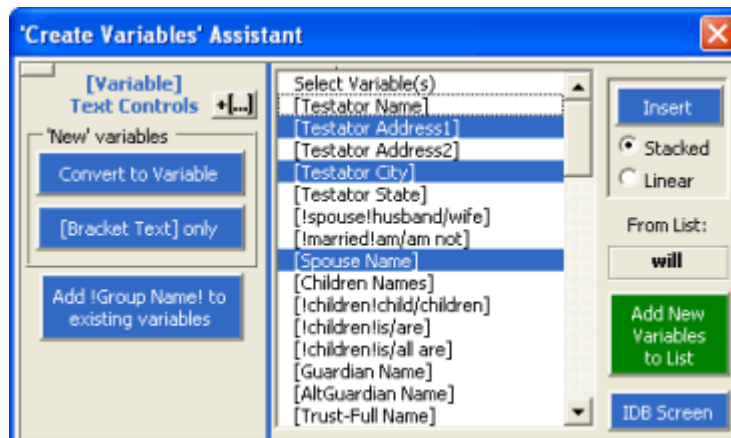
Assuming that you have a mask which contains the appropriate variables:

- 1) Drop down the list at the right side of the CVA that initially reads 'Available Masks.' Select the desired Mask Pathagoras will load the variables into a selectable list. The form will move to the other side of the page, but that is more cosmetic and informative (informing you that you have done something 'big').
- 2) 'Point and click' to move a variable from the list into your document.
- 3) If you had highlighted the text which you want to replace with the selected variable, Pathagoras will present the following screen with options as to how you might want to handle other text identical to the highlighted text. (In the example, the highlighted text was the name "John Q. Doe" and you had selected "Testator Name" from the variables list:



Choose the scope you desire for Pathagoras to make replacements. (Typically, the third choice -- 'entire document'-- is appropriate.

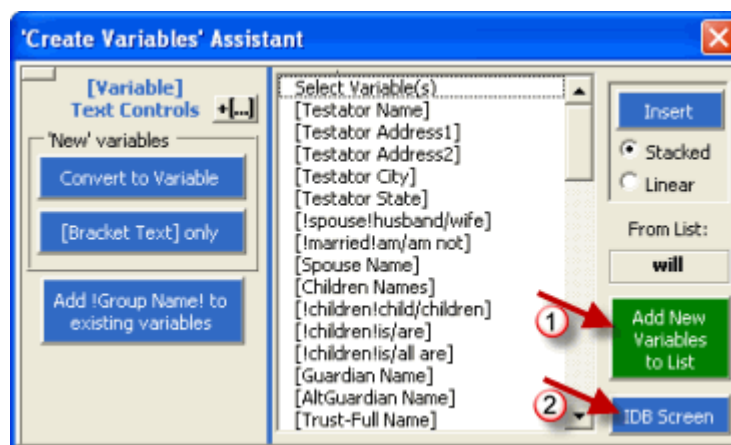
4) Inserting multiple variables.



- a) Normal 'list controls' apply to the display so you need only use Ctrl or Shift plus the left mouse button to select more than one variable. Click Insert.
- b) If 'Stacked' is selected from the options just below the Insert button, an 'enter' will be inserted between each variable. If 'Linear' is selected, a space will be inserted, but the variables will otherwise be inserted on the same line.
- c) You can also 'double click' on a variable to insert it into the document.
- d) If you previously highlighted the text that you want to replace with a variable, and select a single variable from the 'multi-select' screen, you will get the same query screen shown in B. 3. above.

4.4.3 CVA Adding New Variables

If the document you are editing contains variables not yet in the variables list displayed in the Create Variables Assistant, you can add those variables to the list. Simply click the 'Add New Variables to List' button (1).



If you want to add instructions or provide examples for other users to follow as they complete the IDB form, click the Show IDB button (2). This will bring up the classic Instant Database screen where you can edit the variables to your heart's content. See [Making Input Form Entries More Meaningful](#) ²⁶⁷ and [Organizing Input Form Entries](#) ²⁶⁸

4.4.4 CVA Sanitizing Document with +[. . .]

"Sanitizing" your document: An alternative method. Let's assume that you are working on the kind of document described earlier. It is a Will with lot's of personal text. You can quickly sanitize using the methods described above OR you can (more manually) pre-marking the document and use the sanitizer button (marked "+[. . .]") near the top of the Create Variables Assistant screen. The pre-markup involves these steps:

1. Type the characters "+[" in front of the first instance of the text you want to convert, and a "]" at the end.

For example:

Original text: "This will confirm your order of one thousand red widgets. You can expect delivery of your order no later than March 25, 2019. Thank you for ordering your supply of widgets from us."

Marked up text: "This will confirm your order of +[one thousand] +[red] +[widgets]. You can expect delivery of your order no later than +[March 25, 2019]. Thank you for ordering your supply of widgets from us."

Note in the example above that the second instance of 'widgets' is not marked. Pathagoras will change all instances of the bracketed text.

2. Click the 'Sanitizer' button from the Create Variables Assistant. When you click the button, Pathagoras will scan for each marked location and present all that it finds in a 'reverse' Instant Database screen. You provide as 'replacement text' the variable you want to replace the personal text.

4.5 The 'Intelligent A'

If I don't know whether the noun is going to be 'apple' or 'banana', or 'Texas' or 'Illinois', how do I know whether the article preceding it needs to be 'a' or 'an'? And do I really have to choose it each time?

Enter the *Intelligent A*. Simply insert the variable `[**a]` (or `[**a/an]` if you prefer the slightly longer version) in front of the variable that calls for the a or an article. When you have finished making your noun choice via the Instant Database screen, Pathagoras will examine what you have done and automatically insert the proper article choice for you.

Here is an example of a sale made between two non-individual entities. (This example also takes advantage of MultiChoice *Lists*):

THIS CONTRACT OF SALE is made this [DOCUMENT DATE], by [BUYER NAME], [****a**] [ORGANIZATION STATE BUYER:*STATES*] [BUYER ENTITY TYPE] and [SELLER NAME], [****a/an**] [ORGANIZATION STATE SELLER:*STATES*] [SELLER ENTITY TYPE] . . .

If you designate the Buyer Organization state as 'Florida', the 'intelligent a' will become just "a". If you designate the Seller Organization state as 'Alabama', the 'intelligent a' will become "an."

4.6 'And/Or', 'A/K/A' and other slashes

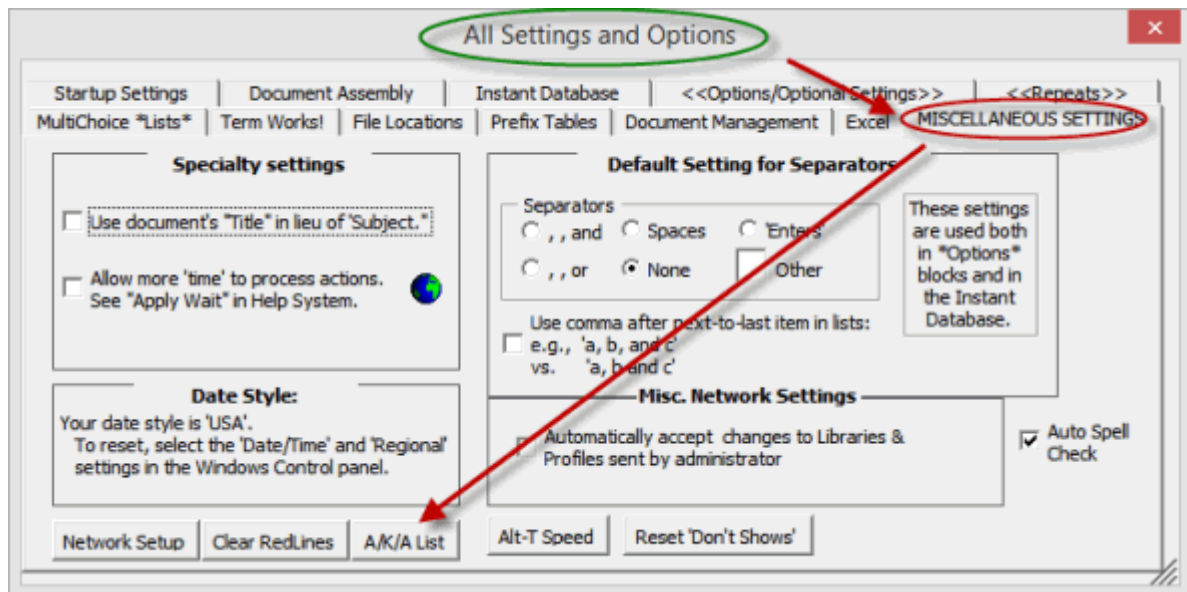
Pathagoras relies heavily upon slashes as separators for Options and for multiple-choice variables. But what if you really want a slash to be seen as a slash and not a separator?


- What if you want 'a/k/a' or 'and/or' to be seen by the program as intact 'words,' not multiple-choices?
- When you are painting your Pathagoras markers, you don't want the slashes in designated terms to be painted?
- The 'Negative Optional' function allows you to set out a block of text to the 'right' of a

slash that will be used if you answer 'No' or 'False' to a preceding Optional block prompt asking 'Do you want to keep this text?'. But what if you know ahead of time that you don't want certain phrases that use slashes to trigger the negative optional choice?

Pathagoras maintains what we call the 'A/K/A List' to effect the above challenges. At installation, the list contains the terms: 'a/k/a', 'f/k/a', 'and/or' and '/100' other snippets, you can augment the 'A/K/A' list to include other text elements that you want 'unpainted.' To do so:

1. Click the Utilities/Settings screen and then select the 'All Settings' button at the top of that screen.
2. Click the Miscellaneous Settings tab.
3. Click the 'A/K/A' button to display the current list of 'ignore' snippets. These snippets will be displayed in a Notepad (plain text) file for easy editing.
4. Add your snippet, following the pattern of the existing entries. Then save and close the text document (do not change its name) and you are all set.



 Pathagoras provides another approach to making sure that a slash is seen as a slash, not a divider. If your multiple-choice variables or your Options text blocks contain 'natural' slash terms, use the separator '/OR' (no quotes, capitals mandatory) as the separator. That tells Pathagoras that the separator in that text block is '/OR' and will ignore simple slashes.

4.7 Reusing Existing Variables

During the early stages of the development of your document assembly system, you likely will be creating and adding variables to your source documents 'manually.' You will identify a proper spot for a variable, type its name and enclose it within brackets. You will repeat the cycle for each variable in the source document. You will 'graduate' to using Pathagoras' "Create Variables Wizard" to create variables in a more global fashion.

But after your system has matured, you may have created Instant Database Records and

Masks which contain the variables that you want to be your 'standard set'. Pathagoras provides even more tools to take advantage of those existing variables. These tools will help you to more rapidly and accurately convert a plain document into a fully 'Pathagorized' document.

4.8 Drop Down List

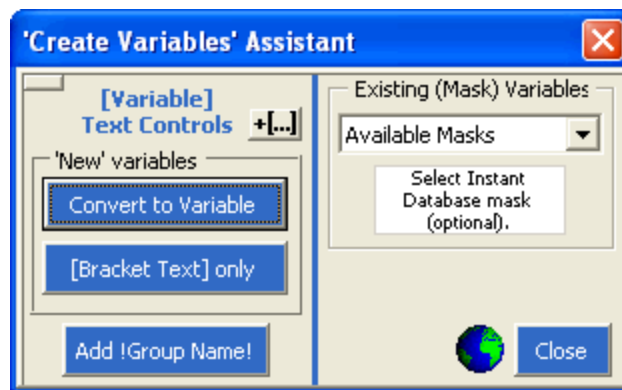
'Pathagorizing' using variables lists.

[Click here](#)³⁷² to learn how to create an 'always on' DropDown List that 'permanently resides' in main Pathagoras tab.

Pathagoras can create a drop down list of existing variables saved in your Instant Database system. This dropdown list works like every other drop down list with which you have worked: 'Point & Click'.


To create the drop down list from the Create Variables Assistant

1. Activate the Create Variables Assistant by pressing <Alt-V>.



2. From the two panel screen that appears, click down the 'Available Masks' drop down list on the right side and select the Mask that contains the variables you wish to insert.

3. Point and click. If you find that you want to insert more than a single variable at a time, check the MultiSelect box that appears in the screen. (Not shown in above image. This option shows only after you have selected the mask.)

 As you probably have assumed, a 'mask' is required for the above described feature to work. Go to part 2 if you do not have any masks but want to use an existing data record as the source of variables. We strongly encourage you to create a mask. It is surprisingly easy to do.

[Read more about 'masks' here.](#)²⁶⁰

To create the drop down list directly from the Instant Database screen:

1. Display the IDB screen (press <Alt-D>, or select Instant Database from the Pathagoras dropdown features menu).

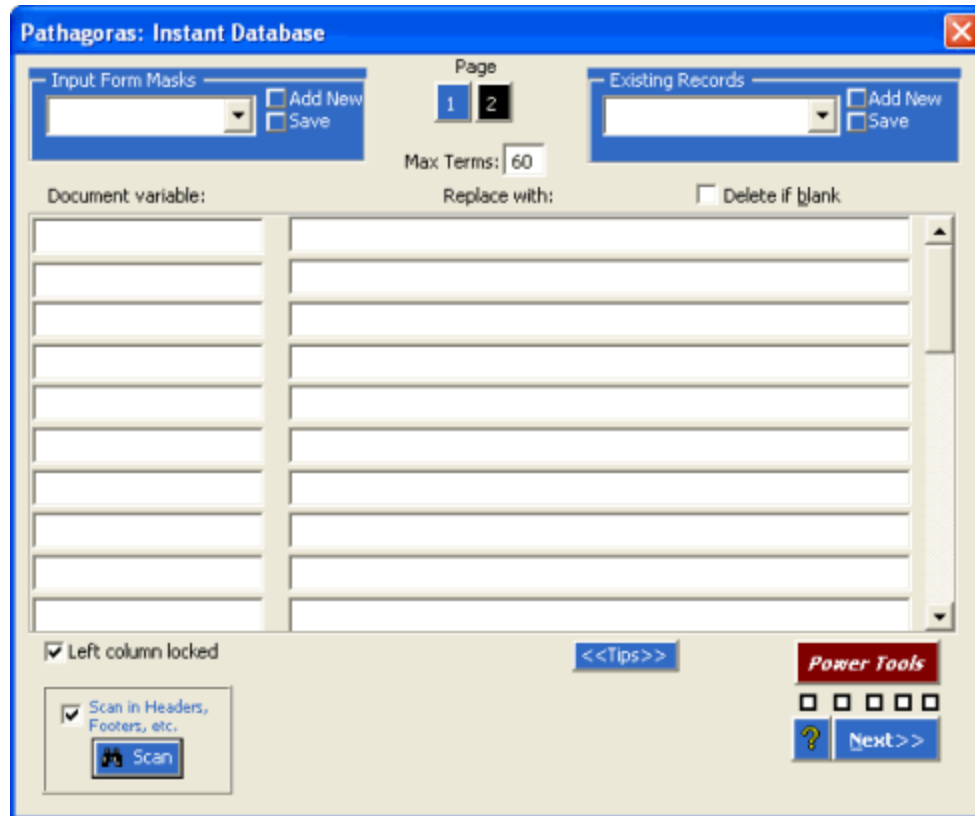


Figure 1. Instant Database Screen (initial view)

2. If you have previously created a mask or have an Instant Database data record which contains the variables that you want to insert into the current document, recall that mask or record by selecting it from the appropriate list in the upper left or upper right corner of the screen. The Instant Database screen will be populated.
3. Alternatively, you can just display any current document containing variables, press the **Scan** button in the lower left side of the screen. Pathagoras will locate each bracketed variable in the current document and list what it finds.
4. Click the Power Tools button at the right side of the IDB screen. Then click the 'Create DropDown list' button in the Editing Tools section. Instantly the items contained in the left side of the Instant Database screen are poured into a drop down list. (If you want the list to appear in alphabetical order, click the alphabetize button at the top of the button column before you create the list.

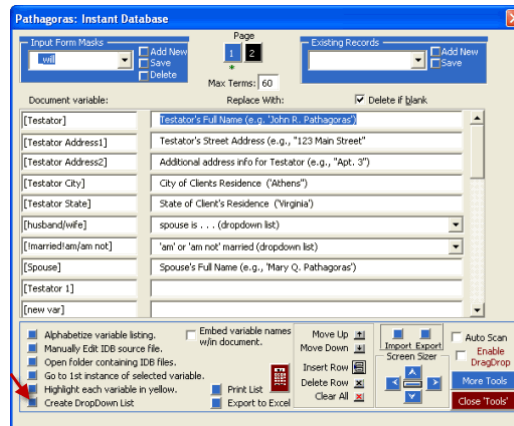


Figure 2. The Instant Database screen after the Settings & Editing Tools button is pressed. To create a dropdown list of the variables in the selected mask, click the Create DropDown List button.

The Instant Database screen will close. There now resides on your screen, in the DropDown List section of your screen, a DropDown List containing the variables.

To insert variables from the list, put your cursor where you want a new variable to be inserted in the source document. Then, in the drop down list on the screen, point to the desired variable and click.

➔ Notes:

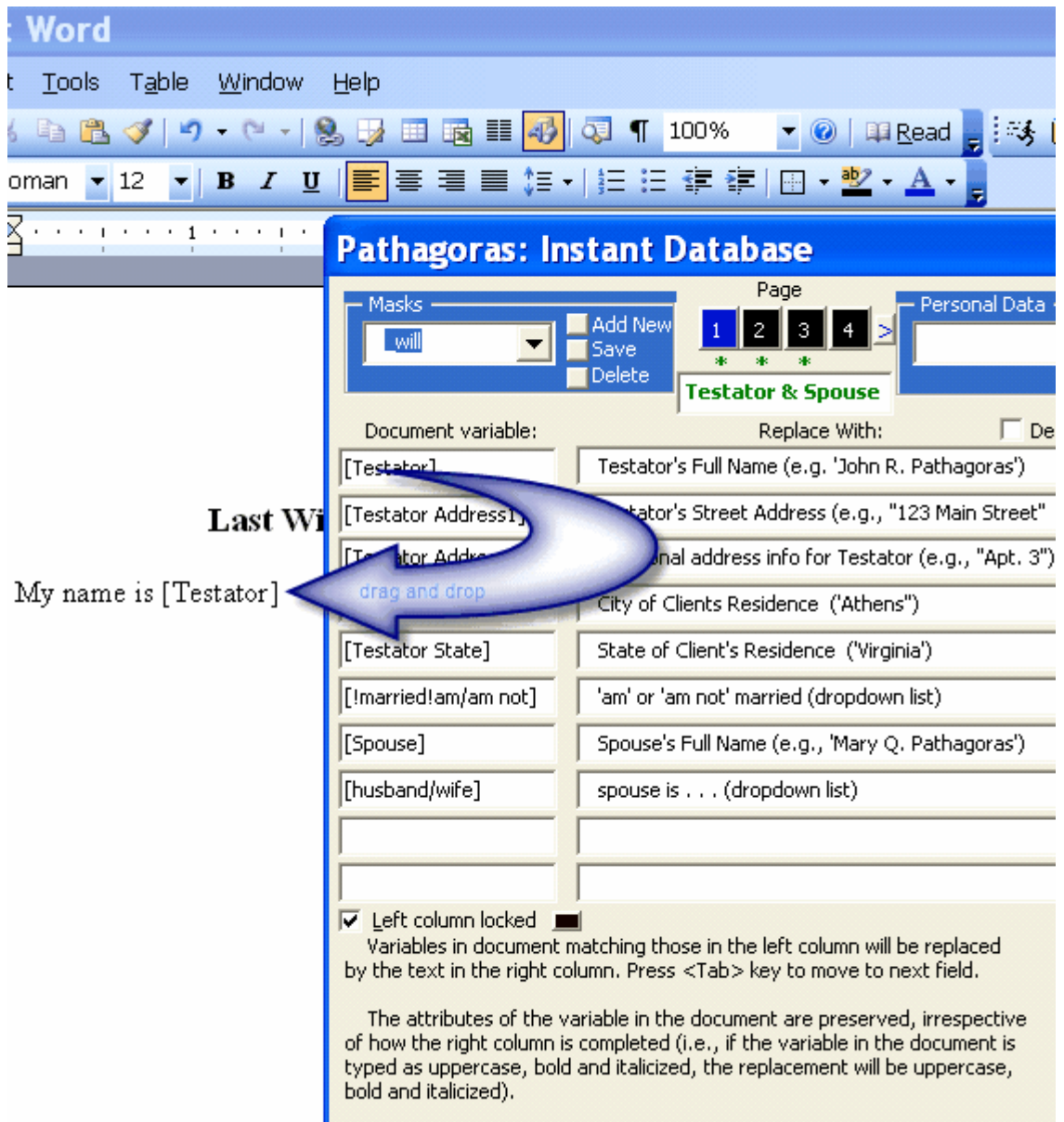
- If you first highlight a word or phrase before clicking on an entry in the drop down list, Pathagoras will
 - replace the highlighted text with the variable and
 - offer to replace all other instances of the highlighted text with the same variable.

4.9 Drag and Drop

'Pathagorizing' using the Instant Database Form

You can use the Instant Database screen directly to move variables from the screen into your document.


1. Display the Instant Database screen, (<Alt-D> will do it.)
2. Select the appropriate Input Form mask from the Masks list (upper left side of the screen). If you aren't using a mask, but the underlying document has variables that you want to use, click the green **Scan** button to populate the screen with variables.
3. Make sure the "Enable Drag and Drop" checkbox (in the the **Power Tools** screen) is checked.
4. Drag and drop variable names from the mask (left column) into your active document. (Click on variable name and hold down left mouse button. Drag into position and release the mouse.)



Drag and Drop Variable from IDB mask into document.

See Also:

[IDB Drop Down Lists](#)  129

[Creating a Mask](#)  262

[Masks](#)  260

[Variable Creation Wizard](#)  122

4.10 Pathagorizing using the IDB screen

While the Instant Database typically is used to *replace* document variables with personal data, the IDB can also be used to *create* variables in an existing document. Here is how.

1. Display a document that you want to 'Pathagorize'. (**Figure 1.**)
2. Put brackets around every name, address or other word that you want to make a variable. (**Figure 2.**)
3. Activate the IDB system (press Alt-D).
4. Press the Scan button to read all of your newly created bracketed 'variables' into the left side of the screen.
5. Type at the right side of the screen the actual variable names you want to appear in the final form document. "[John R. Doe]" (as picked up by the Scan) should become "[Client Name]", "[1234 Oak Avenue]" should become "[Client Address]", etc. (**Figure 3.**)
6. Press the Next button on the Instant Database screen. The 'personal' text (shown in the left column) will be overwritten by the variable text at the right. (**Figure 4.**)

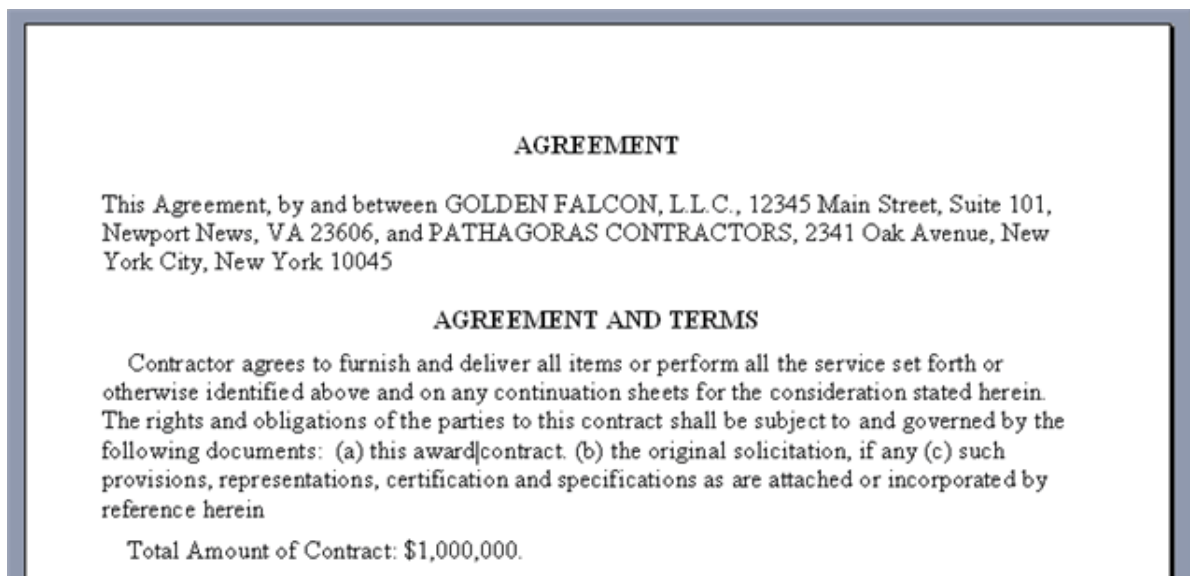


Figure 1. The original (personal) document.

AGREEMENT

This Agreement, by and between [GOLDEN FALCON, L.L.C.], [12345 Main Street, Suite 101], [Newport News, VA 23606], and [PATHAGORAS CONTRACTORS], [2341 Oak Avenue], [New York City, New York 10045]

AGREEMENT AND TERMS

Contractor agrees to furnish and deliver all items or perform all the service set forth or otherwise identified above and on any continuation sheets for the consideration stated herein. The rights and obligations of the parties to this contract shall be subject to and governed by the following documents: (a) this award|contract. (b) the original solicitation, if any (c) such provisions, representations, certification and specifications as are attached or incorporated by reference herein

Total Amount of Contract: [\$1,000,000].

Figure 2. The marked up document.

Pathagoras: Instant Database

Masks: ☐ Add New ☐ Save

Page: 1 2 3 4 >

Personal Data: ☐ Add New ☐ Save

Next >>

Max Terms: 60

Document variable: Replace with: ☒ Delete if blank

[GOLDEN FALCON, L.L.C.] *	[Prime Contractor]
[12345 Main Street, Suite 101] *	[Prime Contractor Address]
[Newport News, VA 23606] *	[Prime Contractor City, ST ZIP]
[PATHAGORAS CONTRACTORS] *	[Sub Contractor]
[2341 Oak Avenue] *	[Sub Contractor Address]
[New York City, New York 10045] *	[Sub Contractor City, ST ZIP]
[\$1,000,000] *	[Amount of Award]

Move Up
Move Down
Insert Row
Delete Row
Clear All

* = IDB term exists in document.

Figure 3. After the document has been scanned to pick up the bracketed text, the left column will be populated with 'pseudo-variables'. Just fill in at the right the 'actual' variables you want to use. Click the Next button at the bottom of the screen to replace 'personal' data with variables.

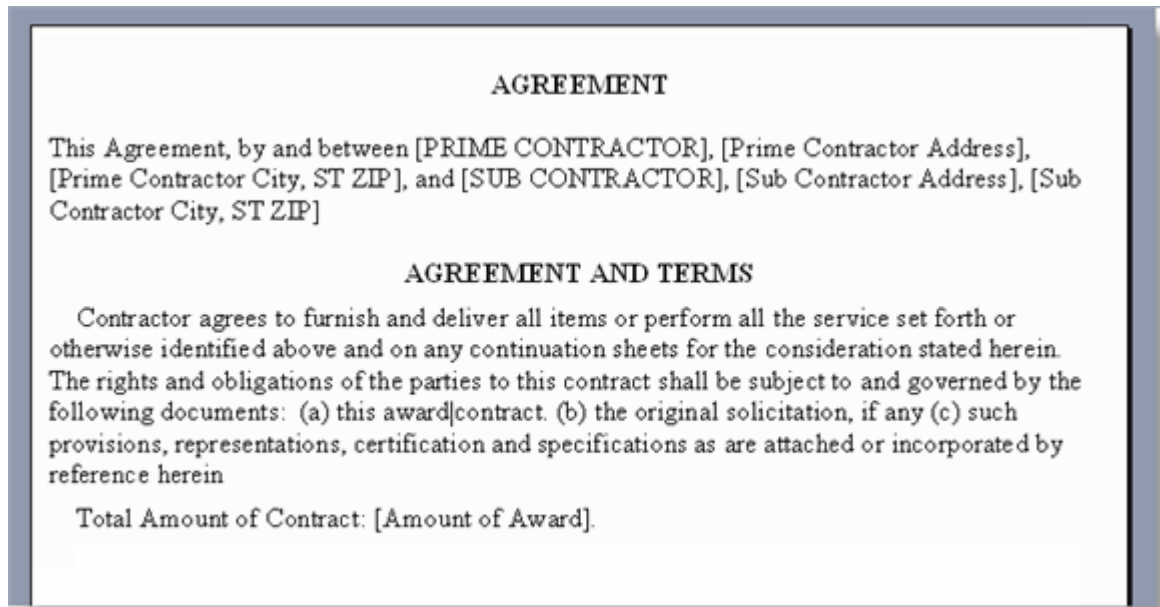


Figure 4. A 'Pathagorized' document!

This technique is an alternative to the [Variable Creation Wizard](#)¹²² and other 'Pathagorizing' techniques discussed elsewhere. Your choice!

4.11 Coverting HotDocs Documents

The Create Variables Assistant can also help you to convert simple variables that reside in documents you may have prepared for HotDocs use. The process is practically identical to that which you would use to convert any other piece of text into a variable. The steps are outlined at [this link](#)⁶⁰⁸.

4.12 Nested Variables

Sometimes the result of a variable needs to be another variable or series of variables. For example, when composing a real estate sales contract, there may be one seller or multiple sellers. Since you don't know at the outset how many, you could have a multiple choice variable that contains more variables. It might look like this:

```
[[Seller]]/[Seller1] and [Seller2]/[Seller1], [Seller2] and [Seller3]]
```

When you make a selection among the multiple choices, Pathagoras will automatically augment the Instant Database screen to include new rows for each of the variables in the selected choice. E.g., if you select the 3rd choice, Pathagoras will add variables '[Seller1]', '[Seller1]' and '[Seller3]' in separate rows immediately beneath the selected variable. (If a variable already exists, it will not be added.)

Here is a slightly longer example based on the above, with some 'real' contract language. Just copy and paste into a document and press Alt-D. Scan and select one of the choices from each to see the 'action:'

```
This Agreement by and between [[Seller]]/[Seller1] and [Seller2]/  
[Seller1], [Seller2] and [Seller3]], hereafter referred to as 'Seller,'
```

and `[[Buyer]/[Buyer1] and [Buyer2]/[Buyer1], [Buyer2] and [Buyer3]]`, hereafter referred to as 'Buyer'.

(Color coding for illustration only to better show nested variables; colors never required.)

Nested Grouped variables: If you desire to assign !groupnames! to the nested multiple choice variables (to maintain number and gender consistency among several variable sets), the above routine will work, but not as smoothly as another approach. We strongly recommend that, instead of nested grouped variable, you use Options (either robust or simple) to accomplish your goal. If you just want the proposed solution, scroll down to **BEST SOLUTION**, below. Nevertheless, we provide examples and suggested language that you could try to maintain a pure 'variables' approach.

Explanation and examples:

The pronoun for multiple sellers is always 'they' or 'them', but for a single seller, the pronouns can be him or her or it (or the possessive 'his, hers or its').

If grouped, the variables might look like this:

```
[!S!Seller/Sellers]
```

```
[!S!his/hers/its/their]
```

Of course, for the grouping-with-multiple-choice-variables to work, the counts (and relative positioning) must be consistent. Here, they just don't 'match up'. There are two choices in the first and 4 in the second.

Here are couple of possible solutions.

- You can add additional choices to the first multiple choice variable. That way, each has the same number of choices:

```
[!S!Seller/Seller/Seller/Sellers]
```

```
[!S!his/hers/its/their]
```

There are several problems with this approach. First of all, it takes up a lot of 'real estate.' The longer a variable is, the more confusing it is to the end users. (This is one of the drawbacks of a 'plain-text' based system, and is one that most users are willing to accept for the myriad of other advantages, but we do not want to ignore the point.)

The second problem is that you don't necessarily know which 'Seller' will return which gender choice. If the second variable is answered first, it works out just fine. However, but unless you know from experience which one to click, you may be confused (or worse, may have confused your end user). (Of course, you can eliminate much of the confusion by creating a [Mask](#)²⁶⁰, or using the 'BEST SOLUTION' as shown below.)

- You can nest the choices for each variable as additional variables.

```
[!S!Seller/Sellers]
```

```
[!S![his/hers/its]/their]
```


Here is a bit more elaborate nesting, listing various combinations of likely parties to a contract:


```
[!S![Seller]/[Seller1] and [Seller2]/[Seller1], [Seller2], and [Seller3]]
```


```
[!B![Buyer]/[Buyer1] and [Buyer2]]
```


```
[!s![his/hers/its]/their/their]
```


[!b![his/hers/its]/their]

 Note also that there must be an identical number of choices for each of the groups. This is illustrated in the very last example where 'their' is repeated a sufficient number of times to match the number choices provided in the other group members.

 As you can observe, the result of selecting the first choice of either group will be another multiple choice variable in the document. A second 'scan' of the document will be needed to complete it. Further, the value set in the second scan will not be saved along with the other values. This likely is not a major drawback, but it is something of which you should be aware.

 Keep in mind the 'top-to-bottom' order in which Pathagoras processes variables. A variable that is nested in another variable may be processed away if the value for that variable 'blank' and you have set the 'Delete if Blank' switch to on.

➔ With the last two  points in mind, it may be (*and probably is*) better to create <<*Options/Optional*>> or {Simple Options/Optional} text blocks to handle the placement of the 'final' variables. <<*Options/Optional*>> and {Simple Options/Optional} blocks are processed first (and automatically) after the document is assembled. That is when decisions regarding numbers of "major players" (parties, buyers, sellers, etc.) need to be made anyway. When properly used, <<*Options/Optional*>> blocks will result in the appropriate variables (and only the appropriate variables) being present when you begin replacing them with the Instant Database routine. Further, the appropriate 'other' nouns, verbs and pronouns are also present. (You can copy and paste the following examples into a Word document and 'process' it to see the action described.)

BEST SOLUTION:

Here is the way you could handle the "Seller" listings above using <<*Options*>> instead of nested variables (remember--the colors in the below examples are for illustration only. They are not required.):

```
<<*Options*!S!One Seller/Two Sellers/Three Sellers*[Seller]/[Seller1] and
[ Seller2]/[Seller1], [Seller2] and [Seller3]>>
. . . (blah blah blah) . . . but if <<*Options*!S![his/hers/its]/their/
their>> . . . (blah blah blah).
```

OR

Here is how to use {Simple Options} instead of nested variables. (Note, Simple Options does not allow 'questions' and requires the "/OR" as a separator:

```
{!S![Seller]/OR[ Seller1] and [ Seller2]/OR[ Seller1], [Seller2] and [Seller3]}
. . . (blah blah blah) . . . but if {!S![his/hers/its]/ORtheir/
ORtheir} . . . (blah blah blah).
```

4.13 'Emphasis' (bold, ital, etc.)

The attributes of the variable in the *source document* controls the look of the replacement text when it replaces the variable. So,

- if the variable in the *source document* is ALL CAPS, the variable will be replaced by ALL CAPS characters, even if the operator types the replacement text in lower case letters.

- if the variable in the *source document* is **BOLD** and *ITALICIZED*, when the variable is replaced, it will appear with **BOLD** and *ITALICIZED* characters.
- The color or emphasis attributes of the variable in the document are always preserved, irrespective of the style of the replacement text. If the variable is blue, red, green, chartreuse or any other color, it will retain the assigned color.

This feature that the replacement text inherits the formatting of the text it replaces allows the same variable (but with different emphasis attributes) to appear in different locations of the same source document. You only have to provide a single replacement value for the variable.

As frequently applies in legal documents, a variable might appear in ALL CAPS, **Bold** and *italicized* (“***[NAME OF CLIENT]***”) in one place in the document (such as the title) and Upper and Lower Case, and no emphasis (“[Name of Client]”) in another.

There is a single exception to the above: If you type the replacement text in ALL CAPS, Pathagoras assumes that you mean it, and that you intend all replacements throughout the document to be in ALL CAPS. Therefore, when you type replacement text into the right column of the Instant Database screen, unless you want all caps, you should type the replacement text in regular upper and lower case style ('John Q. Doe', not 'JOHN Q. DOE').

Here is an example of this 'replacement protocol in a 'Will'. We are replacing [Testator] with 'John Q. Pathagoras' and [City of Residence] with 'Hampton'. Note in particular the appearance of [TESTATOR] in the title lines and [Testator] in the body.

**LAST WILL AND TESTAMENT
OF
[TESTATOR]**

I, [Testator], presently of [City of Residence], Virginia, do make,
publish and declare this to be my Last Will and Testament, hereby revoking
all Wills and codicils heretofore made by me.

**LAST WILL AND TESTAMENT
OF
JOHN Q. PATHAGORAS**

I, John Q. Pathagoras, presently of Hampton, Virginia, do make,
publish and declare this to be my Last Will and Testament, hereby revoking
all Wills and codicils heretofore made by me.

Bottom line: do not worry about capitalization, bold, italics or underlines as you are completing the IDB screen. Type text in a normal fashion. Typically you will type "Upper And Lower" case for names and titles, lower case for most everything else.

Here are a few examples of how a variety of variables will be replaced with a variety of values:

<u>Document Variable</u>	<u>Replacement Text</u>	<u>Result after Replacement</u>
[Color of Paint]	Navy Blue	Navy Blue
[color of paint]	Navy Blue	navy blue
[Color Of Paint]	Navy blue	Navy blue*
[color of paint]	Navy blue	navy blue
[COLOR OF PAINT]	Navy blue	NAVY BLUE
[Color Of Paint]	navy blue	Navy blue*
[color of paint]	navy blue	navy blue
[COLOR OF PAINT]	navy blue	NAVY BLUE
[color of paint]	NAVY BLUE	NAVY BLUE

*This is a bit of an aberration. Because the first letter of the second and third words of the variable is capitalized, the more logical replacement would be 'Navy Blue.' Be aware of this behavior.

➔ **Note:** If you use a **!groupname!** as part of the variable name, the above rules still pertain. You must, therefore, make sure that the **!groupname!** reflects the case you want carried forward when the variable is replaced.

[!Client!he/she/it] will return 'He' or 'She' or 'It', but [!client!he/she/it] will return 'he' or 'she' or 'it'. ➔

Forcing an exception: 'Exact Replacement'

In some documents, you don't want the sophisticated replacement protocol offered by Word to control. You want the replacement text to be identical to that typed in as the 'replacement text.' That is possible. To enable 'Exact Replacement,' click the 3rd element of the Advanced Array checkboxes just above the Next button.



Advanced Array. Click 3rd button to replace text exactly as typed.

Don't worry if you cannot remember the "third checkbox." Simply hover the mouse over any box in the advanced array. A prompt will appear describing the function of the box.

[Click here to read more about the Advanced Array.](#) ²¹⁸

4.14 Date, Formula and 'File Name' replacements

Typically a variable will be replaced with a 'value'. For example, the variable '[Client Name]' might be replaced with "Ulysses S. Grant" and '[Sales Price]' might be replaced with "\$1,000,000"

But Pathagoras requires no such limitations. A variable can be replaced with
a file name or
a folder name so that you dynamically select a file in the folder.

Pathagoras can be replaced with the value of another field.

A Pathagoras variable can be replaced with a formula based on the value(s) of other fields.

A Pathagoras variable can be replaced with a date chosen using a build in calendar.

4.15 Limitations on Variables

There are only a few limitations:

Length:

This is not a true limitation, but there are practical limitations. See [this page](#) ¹⁹⁵ of the Manual for more information.

Content:

A variable can contain any characters you desire. However, but be away that certain characters, such as quotation marks and apostrophes can be represented in Word in two distinct ways using slightly different characters.

For example, compare 'curly' quotation marks “ ” vs. straight quotation marks " ".

Likewise, compare curley single quotes: ‘ ’ vs. ' and the curly apostrophe ’ vs. the straight apostrophe '.

When processed within a variable, a curly quote or curly apostrophe is returned as 'different' from the same word with a straight quote. Therefore, [Client’s Name] is not the same as [Client's Name].

Because of this fact, you should refrain from using quotes and apostrophes as part of a variable name. As a practical matter, [Client Name] is just as meaningful to the end user as is [Client's Name] and it avoids the curly vs. straight issue.

4.16 <<*Options/Optional*>> Text Blocks

<<*Options/Optional*. . .>> text blocks are specially marked sections of text that allow the end user to choose whether all or portions of the text block will remain in the final document.

Two types of optional text blocks are used in Pathagoras. They are both created with, and denoted by, plain-text boundary markers “<<” and “>>” placed around the subject text. (*Pathagoras provides a 'simplified' version of both of these blocks. They are discussed starting at [this link](#)* ¹⁶⁴.)

1. **“Optional” text:** This is pure ‘optional’ (take it or delete it) text.’ It is sometimes called ‘conditional’ text. At document assembly time, the program will highlight the text, pause and ask “Do you want to keep this text?”. The user need only respond “Yes” or “No” to tell Pathagoras whether the text block should be retained in the final document.

```
<<*Optional*The widgets you have ordered are not currently in stock. We
will ship them as soon as possible. If we have not shipped within 5 days of
this date, you will have the option to cancel the order.>>
```

2. **“Options” text:** This block ‘type’ allows the user to select among several choices. Each choice is separated from the others by a simple, plain text slash (“/”). At document assembly time, the choice are presented to the user on a selection screen. The user selects one or more of the displayed choices.

Beginning with version 2013.6, you can use rows in a table to separate the various choices. See this page for examples.

```
<<*Options*Per your request, the widgets will be shipped by Federal
Express. We will bill you for the extra cost of shipping./Per your request, we
will send the widgets by standard ground transport. This may take 3 to 5
additional days./As per your request, we will hold the widgets for pickup by
your courier./The widgets you have ordered are not currently in stock. We
will ship them as soon as possible. If we have not shipped within 5 days of
this date, you will have the option to cancel the order.>>
```

When Pathagoras encounters the optional text block (#1 above), it will highlight the text in the document and ask if you want to keep it (Figure 1).

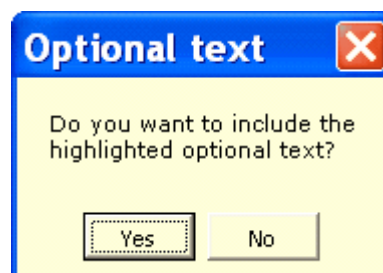


Figure 1 Optional Text dialog.

If you select <Yes>, the boundary markers are removed
and the text remains in the document.

If you choose <No>, the entire text block is deleted from the document.

When Pathagoras encounters the options block (#2 above), it will parse out the individual

choices and display them onto buttons on a selection screen. (If the text is too long to fit, only the first 200 or so characters of the particular option will display.) Checkboxes are shown at the left of each choice so that you can choose more than one option, if desired. If you want just a single choice, click on the 'bar' containing that choice.

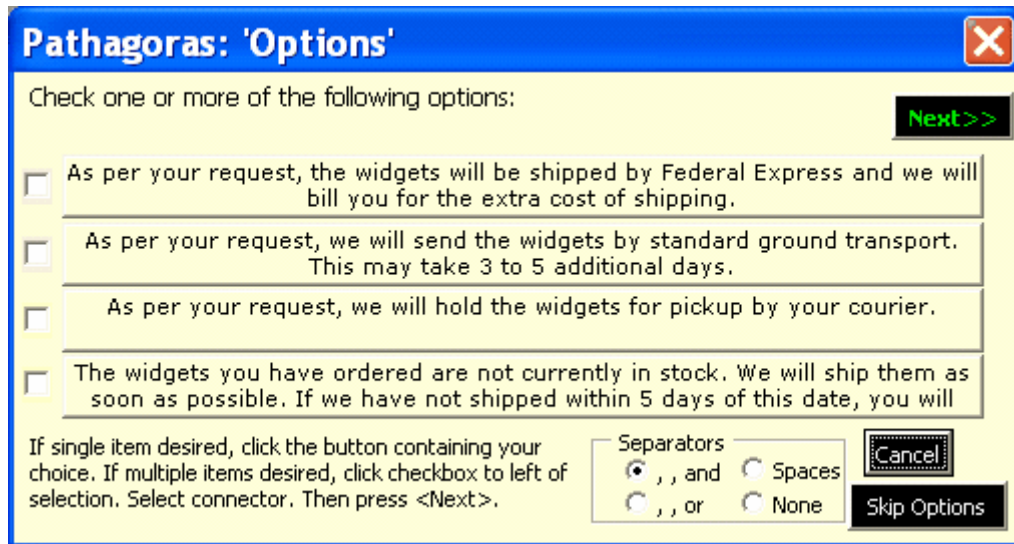


Figure 2 Options block dialog.

Note that the actual option text is provided (subject to space constraints.)

Notes:

- Creating an <<*Options*>> or <<*Optional*>> text block really is no more difficult than what is described above. It is all plain text construction. No fields, no codes. Just remember that the 'administrative' section of an Options or Optional text block must end with an asterisk.
- Instead of processing <<*Options*>> or <<*Optional*>> text blocks 'one at a time' (as the above examples and screen shots suggest) you can enable [Document Logic](#)⁴⁹², creating an Interview Screen and assigning values for all blocks from a single input screen.
- We encourage you to liberally use <<*Options*>> and <<*Optional*>> text blocks. But it is possible to 'over-populate' a document with Optional text. By creating such a document, you will end up with a long, complex looking document that may actually 'scare' the end-user and impede the document assembly process.

If you have inserted dozens of <<*Optional*>> text blocks inside your document, we would suggest that you consider creating individual 'selectable' clauses from the same text. Then, assemble the document from those separate clauses via the Clause Selection Screen (or insert them via a DropDown List). You will likely find this process more efficient and significantly faster. See this discussion: [Optional Text vs. Individual Clauses](#)⁶³¹

- You can have only nine *Options* in an <<*Options*>> block. There are practical reasons for this, not the least of which is the amount of screen 'real estate' that is needed to display the various options in the Dialog (Figure 2). There are two 'work arounds' to this limit.

1. The first is to break the options into two or sets of (up to) nine choices, and create a 'super' options block encompassing the (nested) others. The 'super' level <<*Options*>> will be presented first. Selecting a choice from that level brings up the (smaller) sub-group.
 2. If each of the options is relatively short (a few hundred characters or so), create a [MultiChoice *List*](#)⁴⁵⁸ of the various options. This routine is easy to implement and will solve most of the 'problems' caused by the limit of 'nine.' The options (and there could be dozens of them in the MultiChoice *List*) are referenced in the source text in this remarkably compact fashion: <<*Options**mylist*>>
- **Administrative Text:** The text to the *left* of the last asterisk in an <<*Options*>> or <<*Optional*>> text block is referred to as the 'administrative text', as distinguished from the actual choices found to the *right* of that last asterisk.) We talk about 'administrative text' in other sections of this Manual.
 - **Substantive Text:** The text to the right of the 'administrative text' in an <<*Options*>> block is the text that will remain (if chosen) in the final document. This text can be 'real' text (such as was shown in the above examples) or the text can be *references* to other documents that you want Pathagoras to find and insert. So, an <<*Options*>> block can read like this:

```
<<*Options*As per your request, the widgets will be shipped by Federal
Express. We will bill you for the extra cost of shipping./As per your
request, we will send the widgets by standard ground transport. This may take
3 to 5 additional days./As per your request, we will hold the widgets for
pickup by your courier./The widgets you have ordered are not currently in
stock. We will ship them as soon as possible. If we have not shipped within 5
days of this date, you will have the option to cancel the order.>>
```

or the same block can read like this:

```
<<*Options*<<shp101>>/<<shp102>>/<<shp103>>>>
```

which shp101,102 and 103 point to text stored as a document in a folder or glossary. The 'double angle brackets' signal Pathagoras to make the call to the appropriate document. See ['Calls to Other Documents'](#)¹⁰⁴ for more information and more examples.

- **Tables and <<*Options/Optional*>> blocks:** <<*Options/Optional*>> text blocks work well within and without tables. The only restriction is that the block cannot span more than a single cell of a table. This includes starting outside of a table and ending within a table or starting within a table and ending outside a table.
- An <<*Options/Optional*>> text block can encompass an entire table. The opening "<<" marker must simply be somewhere above the table and the closing ">>" marker must be below it.
- **Where to place the slash?** Typically, the placement of the slash between choices is easy. When words or sentences are being separated, the slash goes after the last character of the preceding choice. But when the choices are 'paragraphs long', a little experimentation and trial and error might be needed. The logic built into Pathagoras as to how to process the keeping/elimination of selected/discarded text is complex. Factors such as paragraph indentions, centering, automatic paragraph numbering, and other style considerations makes it impossible to articulate a hard and fast rule regarding where to

place the slash. Just perform a couple of test runs using the [Process button](#)¹⁶⁰ to see if the result is what you expect. Be sure to test all options.

While <<*Options/Optional*>> text blocks are easy to create manually following the instructions provided above, you can easily build them automatically using the [Create Options Assistant](#)¹⁵⁵.

Any of the text blocks shown above can be copied and pasted into any Word document for testing and experimentation. To test the action of any <<*Options/Optional*>> routine, place the cursor in your Word document immediately to the right of the closing bracket and press Alt-G.

Click here: www.pathagoras.com/sample for a collection of *Options/Optional* text blocks samples which can be copied into a Word document and tested.

See also:

[Create <<*Options*>> Assistant](#)¹⁵⁵

[Administrative Text](#)¹⁶²

[Single \(Radio button\) vs. Multiple Selections \(Checkboxes\)](#)¹⁴⁸


[Structure Checker](#)¹⁷³

[Suppress processing of <<*text blocks*>>](#)¹⁵⁸

[MultiChoice *Lists*](#)⁴⁵⁸

4.16.1 'Optional' Text ('Prompts')

The default action of Pathagoras when it encounters <<*Optional*>> text is to present the entire text block to the user, with the question "Keep? Yes or No." With just a little more effort, you can provide a more meaningful "keep or delete" question to which the user can respond. To do so, add a prompt immediately after the word *Optional*, and note the end of the prompt with an "*".

 All of the text to the left of that third asterisk is called '**Administrative Text**'. The administrative text is never part of the substantive text that you are choosing to keep or delete, but does help you and the end user in making the decision whether to keep it or delete it.

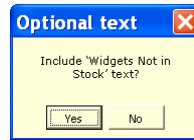
Structure:

<<*Optional*(prompt)* . . .>>

Example:

```
<<*Optional*Include 'Widgets Not in Stock' text?*The widgets you have
ordered are not currently in stock. We will ship them as soon as possible. If
we have not shipped within 5 days of this date, you will have the option to
cancel the order.>>
```

When encountered during document assembly, the prompt between the asterisks are presented instead of the default question shown in the previous page.



Optional Text (displaying user provided question.)

Notes:

- Make sure that the prompt is answerable 'Yes' or 'No,' with the 'Yes' answer resulting in inclusion of the text, and 'No' resulting in deletion of the text. (Note--the colors in the above and the following examples are for illustration purposes only.)
- If you decide that the question needs reworking, simply open the source document and edit the question. Since there is no hidden coding or fields, there is never an impediment to perfecting your source documents.
- In order to enable the Interview Wizard to process Optional text, it must be assigned either a !GroupName! or a Prompt.
- <<*Optional*>> text *in tables*: While it is possible to use <<*Optional*>> text blocks within tables, you cannot cross table cells. That is the opening "<<" marker cannot be in one cell and the closing ">>" marker in another. If you must 'cross' cells, create two separate blocks. If assign a common 'group' name to each block, they will process as if one. See [Groups](#)¹¹⁹ below. You may also wish to consider 'paragraph assembly.' The entire block of 'optional' text is saved as a single clause that you can select from either the [Clause Selection Screen](#)⁴² (along with others that can make up the entire document) or from a [DropDown List](#)³⁵⁶.
- The '**Negative Optional**': You can provide a 'false' value to optional text. The 'false' value will be selected if you choose "No" or "False" to the question presented. To provide the 'false' part, simple type "/NEGOPT" (no quotes) at the end of the Optional text block and type the language you want to be included in the document if the user provides a "No" or : "False" (If the user answers True or Yes, the first part remains and the false part is deleted, If the user answers 'False' or 'No', the second part remains and the 'true' part is deleted. (The action is similar to that of an <<*Options*>> block where two choices are presented, but simpler in structure.)
- **Hover-Over text**: In addition to 'prompt' text that will appear on the pop-up screen which asks "Keep or Delete", you can provide up to 256 more characters of 'hover over' text. This hover-over text will appear when the user moves the cursor over the selection. It will display only during the time the cursor is 'hovering over' the selection. To add hover-over text, simple type a '+' sign at the end of the prompt text and then type the hover-over text. Make sure the hover-over text is within the administrative section of Optional block (i.e., before the third asterisk). E.g.,

```
<<*Optional*Include 'Widgets Not in Stock' text?+Use when product out of
stock and are not expected to be replenished within 3 days of order*The
widgets you have ordered are not currently in stock. We will ship them as
soon as possible. If we have not shipped within 5 days of this date, you will
have the option to cancel the order.>>
```

See also:

[Create <<*Options*>> Assistant](#)  155

[Structure Checker](#)  173

4.16.2 'Options' Text ('Prompts')

<<*Options*>> Text Prompts


The default action of Pathagoras when it encounters <<*Options*>> text is to present the entire text block to the user. Therefore, entirety of the individual options will be presented on the (up to 9) buttons of the selection screen.

This may be satisfactory when the options are entirely self-explanatory, but such is not always the case. With just a little more typing, you can provide succinct and possibly more meaningful 'prompts' (or questions) which will appear as the text on the buttons. The user can respond to the prompts in order to make a selection.

To add prompts to your Options text, list them immediately after the word *Options*. Separate each prompt with a slash (making sure that you have as many prompts as you have options), and close the list of prompts with an "*"

Structure:

<<*Options*(prompt1/prompt2)(etc.)* . . .>>

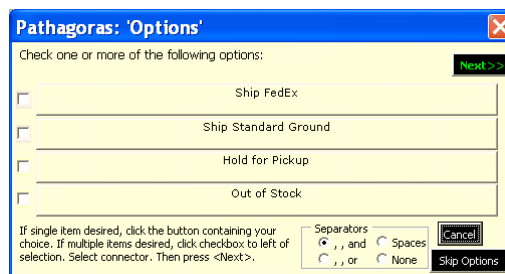
 The text to the left of the last asterisk is referred to as the 'administrative' text, as distinguished from the actual choices found to the right of that last asterisk.)

Example:

```
<<*Options*Ship FedEx/Ship Standard Ground/Hold for Pickup/Out of
Stock*As per your request, the widgets will be shipped by Federal Express
and we will bill you for the extra cost of shipping./As per your request, we
will send the widgets by standard ground transport. This may take 3 to 5
additional days./As per your request, we will hold the widgets for pickup by
your courier./The widgets you have ordered are not currently in stock. We
will ship them as soon as possible. If we have not shipped within 5 days of
this date, you will have the option to cancel the order.>>
```

When encountered during document assembly, the prompts between the asterisks are presented instead of the actual series of options. (See example of 'no prompts' on the previous page).

The above example will yield the result shown in the figure below:



Options block, with descriptions provided.

i NOTES -- "and/or", "he/she" (etc.) and fractions: By default, Pathagoras uses the forward slash '/' character to determine the boundaries of each option in an <<*Options*>> block. However, if you use the forward slash as 'normal text' within an <<*Options*>> text block (e.g., a fraction like '1/2' or a non-variable phrase such as 'he/she' or 'and/or'), you must use "/OR" as the separator within that particular block. Consider the following.

```
<<*Options*Chocolate/Vanilla/Mixed: 1/2 chocolate and 1/2 vanilla.>>
```

In the above example, Pathagoras cannot tell where the 'real' choices end. It will see the above block as 5 choices ("Chocolate", "Vanilla", "Mixed: 1", "2 chocolate and 1" and "2 vanilla") and not the intended 3.

As noted above, the workaround requires only that you add the word "OR" to the slash (to create "/OR") to denote the choices. So the above block should read.

```
<<*Options*Chocolate/ORVanilla/ORMixed: 1/2 chocolate and 1/2 vanilla.>>
```

A few more notes:

- "/OR" must be in CAPS.
- If you use shorthand text prompts (described at the very top of this page) to depict each option, you must use "/OR" in the prompt as well as in the <<*Options*>> block body.
- "/OR" is only required when a natural slash exists within the same <<*Options*>> block. It is not needed otherwise.
- While the 'rule' is that you must have as many prompts as you have options, the exception to the rule is that you can have a single prompt as 'introduction text' to an Options block. Therefore, this is a 'legal' construct:

```
<<*Options*Choose your favorite flavor:*Chocolate/Vanilla/Strawberry.>>
```
- <<*Options*>> text *in tables*: <<*Options*>> text blocks work well within a single cell of a table, but you cannot cross table cells. That is, the opening "<<" marker cannot be in one cell and the closing ">>" marker in another. If you must 'cross' cells, consider 'paragraph assembly.' Each block of text comprising an option is saved as a single clause. You can select the appropriate clause from either the [Clause Selection Screen](#)⁴²¹ (along with others that can make up the entire document) or from a [DropDown List](#)³⁵⁶.
- The options to the right of the 'administrative text' can be 'end text' choices (such as chocolate and vanilla) or they can be references to other text, including calls to documents, if enclosed within 'double angle brackets'. See '[Calls to Other Documents](#)'¹⁰⁴ for more information and examples.
- **Hover-Over text:** In addition to 'prompt' text that will appear on each 'choice button' on the screen which presents the various options, you can provide up to 256 more characters of 'hover-over' text for each choice presented. This hover-over text will appear when the user moves the cursor over each choice. It will display only during the time the cursor is 'hovering over' the selection. To add hover-over text, simply type a '+' sign at the end of the prompt text and then type the hover-over text. Make sure the hover-over text is within the administrative section of Options block (i.e., before the third asterisk).

See also:

[Create <<*Options*>> Assistant](#)  155

[Structure Checker](#)  173

[MultiChoice *Lists*](#)  458

[Single \(Radio button\) vs. Multiple Selections \(Checkboxes\)](#)  148

[Calls to Other Documents](#)  104

4.16.3 Radio vs. Multiple Choice

The “Radio Buttons” Option

If you look carefully at the result in the previous examples, you will see checkboxes next to each option and a “Separators” box at the bottom. If you wish to restrict the use to making a single choice, use ‘radio buttons’. Radio buttons are enabled by adding the word “(radio)” (parenthesis required) next to the word 'Options'. (Note that the closing “*” is still required to close out the "Options" designation.)

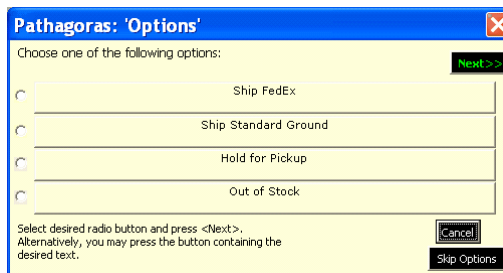
Structure:

```
<<*Options(radio)* . . .
```

Example:

```
<<*Options(radio)*Ship FedEx/Ship Standard Ground/Hold for Pickup/Out of
Stock*As per your request, the widgets will be shipped by Federal Express
and we will bill you for the extra cost of shipping./As per your request, we
will send the widgets by standard ground transport. This may take 3 to 5
additional days./As per your request, we will hold the widgets for pickup by
your courier./The widgets you have ordered are not currently in stock. We
will ship them as soon as possible. If we have not shipped within 5 days of
this date, you will have the option to cancel the order.>>
```

This is the result:



Options presentation with 'radio' buttons. (Compare to above)

4.16.4 Nesting

<<*Options/Optional*>> Text Blocks: Nesting:

Nesting of <<*Options/Optional*>> text block is allowed. As a practical matter it is not advisable to nest beyond two or three levels. This is not because the system itself is limited. Rather this limit is suggested because it gets confusing to the eye to have nestings too deep.

Here is one example of nesting based on the examples used in prior pages. The nest appears

in the last option. (Coloring has been added to the elements of the text block for emphasis. You can also add coloring to your source documents to make editing easier. See [Color Markers](#)⁴⁴⁷):

```
<<*Options(radio)*Ship FedEx/Ship Standard Ground/Hold for Pickup/Out
of Stock*As per your request, the widgets will be shipped by Federal Express
and we will bill you for the extra cost of shipping./As per your request, we
will send the widgets by standard ground transport. This may take 3 to 5
additional days./As per your request, we will hold the widgets for pickup by
your courier./The widgets you have ordered are not currently in stock. We
will ship them as soon as possible. If we have not shipped within 5 days of
this date, you will have the option to cancel the order. When available, the
widgets will be shipped by: <<*Options(radio)*Federal Express./Standard
ground transport./Per your request, we will hold the widgets for pickup by
your courier.>>>>
```

As you are designing <<*Options/Optional*>> text blocks, please note that the text need not be a solid block (as the above examples were). <<*Options/Optional*>> text blocks can be multi-dimensional, just like Word documents can be. You can have single or multiple, paragraphs, add 'character' to the text (color, emphasis, even pictures). You can add any dimension to make the block meet your specific need. (Just a reminder. The various elements of the text block in the below illustration are 'painted' red and blue. These colors are for emphasis only. Colors are not required See [Color Markers](#)⁴⁴⁷.)

<<*Optional*My favorite foods are:

<<*Options*Vegetables

My favorite vegetable is <<*Options(radio)*tomatoes/potatoes/squash/
beans/none of the above>>./

Fruits

My favorite fruit is <<*Options(radio)*peaches/<<*options(radio)
*Macintosh apples/Golden Delicious apples/Granny Smith apples/plain ole
apples>>/plums/bananas>>./

Grains

My favorite grain is <<*Options(radio)*whole wheat/barley/oats/beer>>.

>>

>>


Note: Pathagoras always works on <<*Options/Optional*>> text blocks from the outside in.

Diversion: Copy and paste the above example into a standard Word document. Trigger the processing by pressing <Alt-G> at the very end of the document. Note the actions of the various elements and option styles used, including the use of non-radio options for the top-most question, and radio options for the others.

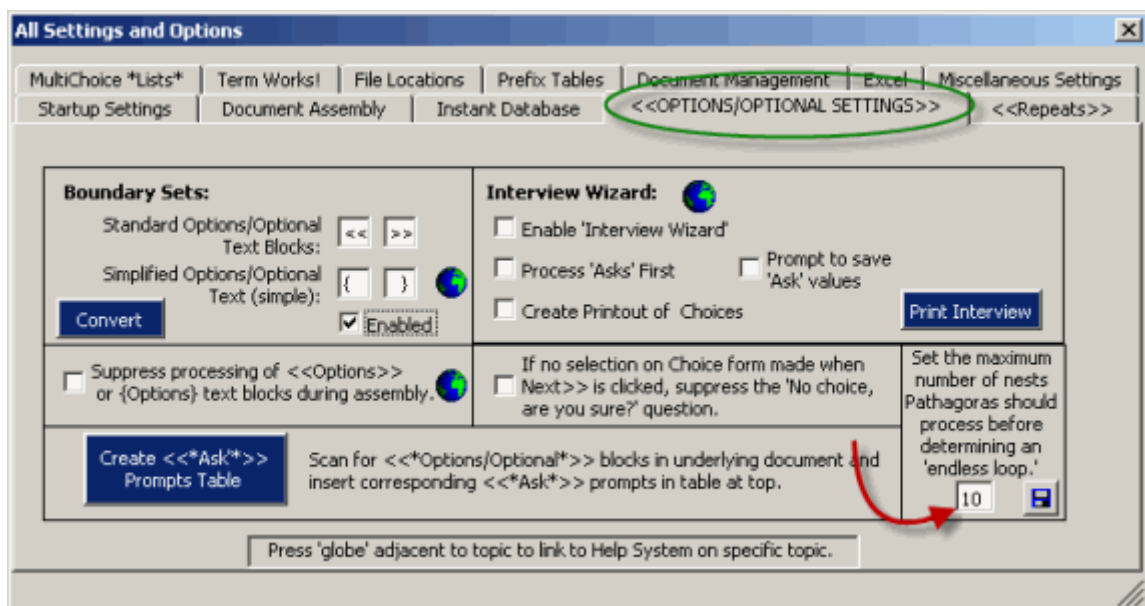


You can nest Options questions to as many levels as you choose. But just because you *can* doesn't mean that you *should*. We suggest no more than two levels of nesting. The problem with nesting is that, the 'deeper' you go, the more difficult the resulting text is to read. Everything in Pathagoras is

on-screen, not in hidden fields. This is a plus when you nest once or twice -- the end user doesn't have much trouble seeing a predicting what will happen if a certain choice is made. But when it gets deeper that that, it can be difficult to read. Concentrate on properly constructed questions. That way, you will not have to go very deep at all.

 As stated immediately above, there is no 'legal limit' to the number of nests you can create, but Pathogoras needs a cap to tell it when to stop looking for closing brackets. An arbitrary number of 5 nestings was selected. (An error message reporting "Hopeless Imbalance" will appear when the cap is reached. (By 'nesting', we refer both to 'deep' nesting -- where every new nest is a 'grand child' of the previous parent, and sibling nests, where all nestings are child of a single parent, on the same level as each other.)

If you receive the message, and know the structure is correct, you will need to increase the cap. To do so, go to the <<Options/Optional Settings>> tab in the All Settings screen and increase the count to 10 (or more, if needed). (See image below.)



4.16.5 Nesting Variables within Options

Variables: Variables of any nature can be placed within an <<*Options/Optional*>> text block. The variables are kept intact while Options/Optional choices are being made. Any variables that remain after the Options/Optional answers are provided can then be processed during the 'final' (Instant Database or GotForms?) phase of the document assembly process.

However, <<Options/Optional*>> text blocks may not effectively be placed within bracketed variables. Because of the processing heirarchy, they will be processed away before you reach the Instant Database phase. If you need to nest <<*Options/Optional*>> blocks, nest them within other <<Options/Optional*>> blocks.

See Also: [Variables](#) 

4.16.6 Groups

Sometimes, the answer to a particular question will lead to a particular answer to a subsequent question. In such cases, rather than responding to each <<*Options/Optional*>> text block, you can create text block 'groupings.' When you answer the question, or select a certain option (or set of options) from the first member of the group, Pathagoras will hunt down other members of the group and select the corresponding choices. E.g.:

Structure:

```
<<*Options*!groupname!(prompt1/prompt2)(etc.)* . . .>>
```

```
<<*Optional*!groupname!(prompt)*. . .>>
```

Note: There must always be two asterisks surrounding 'Options' or 'Optional' (so it will be distinguished from other appearances of these words in your text). If you have added a !group name! or prompts/questions, a final '*' must be used to close that 'administrative' section of the block.

Note: At the risk of repetition ad nauseam, but recognizing that this may be the first time you have seen this, the colors used in the examples are not required. they are for emphasis only. If used in your document, the colors will have no effect on the final product.

Examples:

We suggest that the color for the living room be <<*Options*!colors!*red/yellow/burnt orange>> and the contrasting trim color be <<*Options*!colors!*yellow/purple/royal blue>>.

<<*Optional*!Disclaimer!Include disclaimer language?*We disclaim all liability if you try to shave your head with our chainsaw.>> (blah blah blah . . .) and furthermore <<*Optional*!Disclaimer!we also disclaim responsibility if you try to use our hedge trimmer to trim your fingernails.>>


Notes:

- The group name (the text between the two "!" marks) can be no longer than 30 characters.
- When used with an *Options* block of text, the automatic selection of subsequent choices by Pathagoras typically depends upon the *position* of the first choice in the list, not its answer. So, in the first above example, if 'yellow' (the second option) is selected when the first *options* block is encountered, 'purple' (the second option in the subsequent options *block*) will be selected automatically when it is encountered.

There are two 'exceptions' to this rule:

(1) When you !group! a <<*Repeat* . . .> block with an <<*Options* . . .>> block, the selection of the <<*Options* . . .>> text depends upon whether '0', '1' or '2+' is the response to the *Repeat* question. [This is discussed here](#) ³⁹².

(2) When the value being selected/tested is numeric (e.g, grades, scores, temperatures), and you are using the <<*AskValue*>> command. The <<*AskValue*>> command directs Pathagoras to select a response that falls within various ranges of values. the <<*AskValue*>> command is discussed here.) ⁵⁰³

 When creating second and subsequent members of a group, it is not necessary to have or to repeat any prompt text in the second (third, etc.) members of the same group. We recommend that you include prompt text for the first appearance of a !group!, but not thereafter.

See also:

[Structure Checker](#)  173

[!Groups! and MultiChoice *Lists*](#)  476


4.16.7 "Syntax"


Options/Optional Text:

The anatomy of <<*Options*>> & <<*Optional*>>text blocks.

The setup for <<*Options*/*Optional*>> text blocks is not without its potential complexities. Compared with the setup required by other programs, however, it is undeniably simpler. Because it is plain text, you don't have to fish around for the right field coding and switches. And when you actually see it 'broken down,' it makes even more sense to you.

But precise placement of the various characters that make up the blocks is mandatory. Without the right characters properly placed, Pathagoras will not be able to dissect and process the text. Not to worry too much, however. After the discussion of 'anatomy,' we will show you (1) how to activate a screen which will automate the creation of <<*Options*/*Optional*>> blocks and (2) how to activate a different screen which will check the structure of each <<*Options*/*Optional*>> block in a document.

 The coloring that you will see in the samples below is for emphasis only. No coloring is required.

 Remember that the 'administrative' section (name, group and prompts, if any) of an <<*Options*>> or <<*Optional*>> text block must end with an asterisk.

Simplest <<*Options*>> block (up to 9 choices separated by slashes):

<<*Options*I have one minor child./I have no children./I have __ minor children.**>>**

(Notes: Colors are only for emphasis. Without 'prompts' (discussed below), the actual text of each of the options will appear on a selection form during document assembly process. Don't forget the closing brackets.)

<<*Options*>> block with reference to MultiChoice *List*

<<*Options**children*.**>>**

<<*Options*>> block with "Questions"

<<*Options*Is there one child?/ Are there no children?/Are there two or more children?* I have one minor child to whom I give, devise and bequeath the remainder of my estate./I have no children./I have [number of children] minor children to whom I give, devise and bequeath the remainder of my estate, in equal shares.**>>**

(Notes: The question — which need not actually be a question — is any text set out between the

second and third asterisks. The questions will appear on the selection form during document assembly process.)

<<*Options*>> with groupings:

<<*Options*!Children!* I have one minor child./I have no children./I have __ minor children.>>

<<*Options*!Children!* I give the remainder of my estate to my child./I give the remainder of my estate to the following persons. /I give the remainder of my estate in equal shares to my children, per stirpes.>>

(Notes: The Group name (in this case "Children!") is set out between exclamation marks immediately after the second asterisk. An asterisk after the !groupname! closes out the administrative section of the block. Once a selection of the first !group! encountered has been made, all other options blocks within the same !group! will be 'handled' in the same fashion.)

<<*Options*>> with groupings and prompts:

<<*Options*!Children!Is there one child?/Are there no children?/Are there two or more children?* I have one minor child./I have no children./I have __ minor children.>>

<<*Options*!Children!* I give the remainder of my estate to my child./I give the remainder of my estate to the following persons./I give the remainder of my estate in equal shares to my children, per stirpes.>>

(Notes: Once the first Options group is encountered and question answered, Pathagoras will automatically process the remaining blocks with the same group name. The selection is made using the same *positions* in the list. Because the processing is done automatically, you do not need to repeat the question for subsequent members of the same group.)

Simplest <<*Optional*>>:

<<*Optional*You may also pick up your block of cement from our help desk between the hours of 9 a.m. and 5 p.m.>>

<<*Optional*>> with question:

<<*Optional*Is pickup available?*You may also pick up your block of cement from our help desk between the hours of 9 a.m. and 5 p.m.>>

<<*Optional*>> with groupings and question:

<<*Optional*!pickup!Is pickup available?*You may also pick up your block of cement from our help desk between the hours of 9 a.m. and 5 p.m.>> (body text . . .body text)

<<*Optional*!pickup!*Be sure bring a crane with you.>>

(Notes: Once the first Optional group is encountered and answered, Pathagoras will automatically process the remaining blocks with the same group name. Because the processing is done automatically, you do not need to repeat the question for subsequent members of the same group.)

Linear summary of the <<*Options*>> block elements. Mandatory elements are noted in red:

<<
Options or ***Options(radio)***
!GroupName! (group names allow you to complete all members of the group with the

same level selection made in the first member of the group.)

[QuestionText/QuestionText/etc*](#) (no opening "", just a closing one. Either one question per option or one question, period.)

[Option1/Option 2/Option Number 3/etc.](#)

>>

Use the '(radio)' suffix within the Options tag if you want to restrict the user to just one of the listed choices. The default is to allow the user to select one *or more* of the up to five possible choices.

Remember that the 'administrative' section of an Options or Optional text block must end with an asterisk.

Linear summary of the <<*Optional*>> block elements. Mandatory elements are noted in red:

<<

[*Optional*](#)

[!GroupName!](#) (group names allow you to complete all members of the group with the same level selection made in the first member of the group.)

[QuestionText*](#) (followed by an asterisk to close out the administrative section)

blah blah blah (The text that you want to stay or to be deleted.)

>>

See also:

[Create <<*Options*>> Assistant](#)  155

['Administrative Text'](#)  162

[Coloring Markers](#)  447

[Structure Checker](#)  173

4.16.8 <<*Options*>> using Tables

Beginning with version 2013.6, Pathagoras allows the construction of Options blocks in two fashions:

The 'traditional' method separates each choice with slashes.

But sometimes presenting the data in rows and columns makes for a better presentation. A prompt line (in blue in the following examples) must be prepared to provide the end user with a list of choice that will be displayed to the end user. But slashes between the actual choices are supplanted by the rows in the table.

E.g.

Here is the slate of nominees for the Board of Directors:

Title	Name
-------	------

<<*Options*!Officers!CEO/President/VP/Secretary/Treasurer/COO/CTO/CFO*

Chief Executive Officer	[CEOName]
President	[PresidentName]
Vice President	[VPName]
Secretary	[SecretaryName]
Treasurer	[TreasurerName]
Chief Operating Officer	[COOName]
Chief Technical Officer	[CTOName]
Chief Financial Officer	[CFOName]

>>

<<*Options*FedEx/USPS (First Class)/UPS*

Federal Express	The widgets will be shipped by Federal Express. We will bill you for the extra cost of handling and shipping.
United States Postal Service	We will ship the widgets by standard First Class mail. There will be no additional charges.
UPS	We ship your purchase via United Parcel Service

>>

<<*Options*!Entity!Corp/LLC*

<<*Repeat*!Directors!* _____	
[Director#Name]>>	
<<*Options*Owners?/Manager?*	
<<*Repeat*!Owners!* _____	
[Owner#Name]>>/	

Manager	
>>	

>>

4.16.9 Create <<*Options*>> Assistant

The Create '<<*Options/Optional*>>' text assistant' can help you to properly form the administrative section of each <<*Options/Optional*>> text block. If you are not familiar with how Optional and Options text, be sure to familiarize yourself first with the [<<*Optional*/Options*>> text blocks](#) ¹⁴¹ section of this Manual.

To activate the Assistant, click the “Create <<*Options/Optional>> Ass't” entry in the Pathagoras drop down features list. (You can also press <Alt-O>.) This screen will appear.

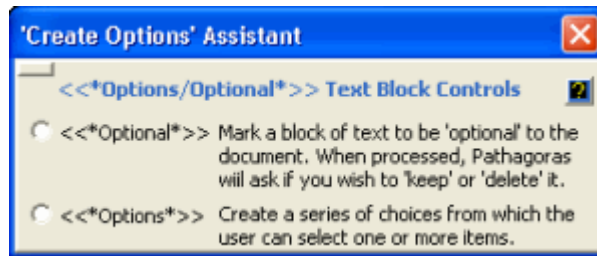


Figure 1. The Initial 'Create Options' Assistant Screen.

There are two distinct types of text blocks with which this Assistant deals.

A. <<*Optional*>>: 'Optional' text is pure 'take or leave it' text.

1. Highlight text you intend to be 'optional'.
2. Display the CVA. Check the <<*Optional*>> control.
3. If you want to add a Group Name (so that all Optional text block throughout the document will be processed together), check that box and provide the Group Name.
4. Click **Next>>**. "<<" and ">>" markers are placed around the selected text and if selected, the Group Name is added as well.

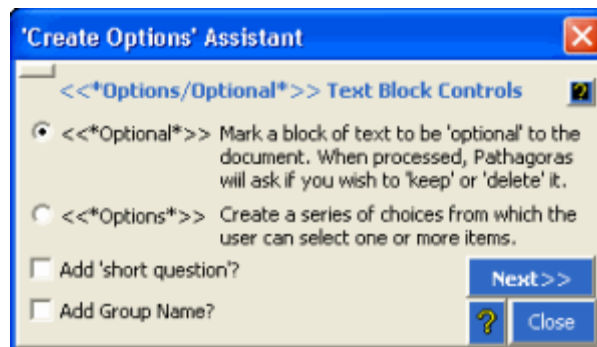


Figure 1. When <<*Optional*>> selected, the screen offers additional choices ('Add Short Question' and 'Add Group Name'), corresponding with advanced features available with <<Optional>> blocks.

B. <<*Options*>>: 'Options' text is that text which provides the user with two or more (up to six) selection from which to choose.

1. Highlight the text which reflects the choices you have composed. (The choices should be separated from each other by forward slashes (/)).
2. Display the CVA. Check the <<*Options*>> control. The menu is expanded to look like Figure 3.
3. If you wish Pathagoras to display 'short questions' in lieu of the entire optional text section when the Options block is encountered during document assembly, check the Add 'short questions' box. Select then number of short question holders you want reserved (same as the number of options you have provided).
4. Select '**Checkboxes**' if you want the user to be able to select more than one of the

options. Select 'Radio **Buttons**' if you want to limit the user to choosing a single option from the ones presented.

5. Press the Next button. Type your 'short questions' in the places indicated.

There you can select whether to add just the opening and closing "<<" and ">>" markers, or place holders for questions you may want to use to guide the user.

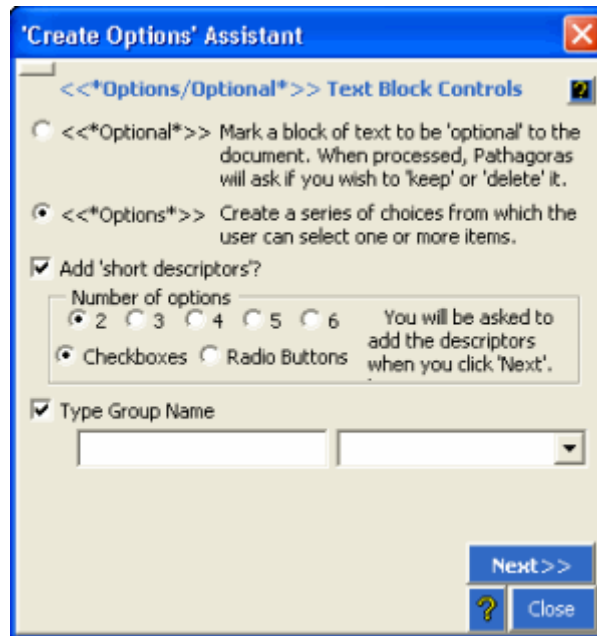


Figure 2. The Expanded Variable Creation Assistant Screen.
(Expanded after the <<*Options*>> radio button was selected.)



Don't be afraid to make a 'mistake' in creating <<*Options/Optional*>> text blocks. After you press an action button, Pathagoras will display an "Undo" button (not shown here). Click it once or twice to take you back to your original text. You can then start over with a different approach.

4.16.10 Options/Optional Text in Headers/Footers

<<*Options*>> text blocks do not (currently) function in headers, footers or footnotes. However, if the value that you want to insert in the header/footer/footnote is calculated in the body of the text, you can duplicate the result by using a bookmark to mark out the options block in the body, and a cross-reference to the result in the header/footer/footnote. Here are the steps.

In the 'source' text (the original letter, form, clause, etc.), place a bookmark around the <<*Options*>> block.

- Highlight the <<*Options*>> block (being careful not to highlight too much or too little).
- Word2003 and before.
 - Click 'Insert' from the toolbar.
 - Click 'Bookmarks'
 - Provide a name for the bookmark (something short, you will only use it one more time).

- Click 'Add'
- Word2007/2010.
 - Click the 'References' tab.
 - Click 'Bookmarks'
 - Provide a name for the bookmark (something short, you will only use it one more time).
 - Click 'Add'

To add the cross-reference, place the insertion point (cursor) at the location (header/footer, etc.) where you want the referenced text to be copied.

- Word2003 and before.
 - Click 'Insert'
 - Click 'Reference' then click Cross-Reference'
 - From the 'Reference Type' dropdown in the resulting screen, indicate that you want to reference a 'Bookmark.' Then from the list of available bookmarks, select the one you just created.
 - Click 'Insert.' That's it. Save your source document and test.
- Word2007/2010.
 - Click the References tab.
 - Click 'Cross-Reference' (which is immediately below 'Bookmarks').
 - From the 'Reference Type' dropdown in the resulting screen, indicate that you want to reference a 'Bookmark.' Then from the list of available bookmarks, select the one you just created.
 - Click 'Insert.' That's it. Save your source document and test.

4.16.11 Simple Options

Pathagoras offers a simpler approach to 'options' and 'optional' text that is appropriate in many situations. Instead of the <<*Options* . . .>> structure used by the 'robust' version, simple 'curley brackets' "{" and "}" are all that are needed to indicate the optional text. [Read more about Simple Options at this link](#) ¹⁶⁴.

4.16.12 Suppress Processing

Processing of double angle bracket (and other text) blocks is automatic unless such processing is 'suppressed.' If you are testing certain actions, or trying to display a certain block of text and don't want to have to respond to the 'keep or delete?' questions inherent with optional text, you will want to suppress the processing of the <<Options/Optional>> blocks

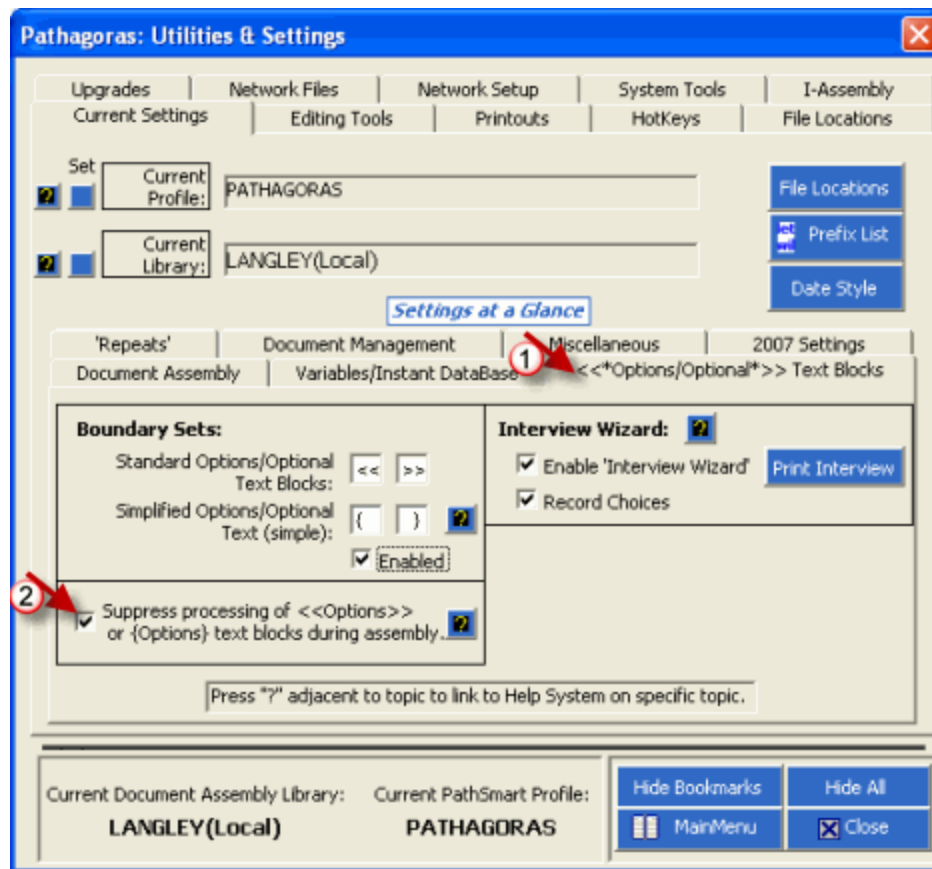
To suppress the processing of <<Options/Optional>> text blocks that are called from DropDown Lists, press the "Suppress" toggle button at the end of a DropDown List row.




Click the Suppress toggle at the end of a row of DropDown Lists to shut down the processing of <<text blocks>>.

In Word2007/2010, the Process/Suppress toggle button is located to the left of the DropDown Lists, underneath the New Doc/Insert toggle.

To **suppress** processing of <<Options/Optional>> called up via other document assembly methods, check the "Suppress *Options* & <<Text>> Block Processing' box on the Current Settings page of the [Utilities & Settings](#) ⁵⁷² screen.



To temporarily suppress text processing by Pathagoras, check the <Suppress> button on the Utilities/Settings Screen (<<Options/Optional>> tab).

 If and when you later decide to process the <<Options/Optional>> text, you have several methods at your disposal:

Option 1: Click the Pathagoras dropdown features menu. Select "Process" and then click the button labeled 'Process Page.' (This is the 'cleanest' way. But the next two options work equally well.)

Option 2: Place the cursor to the immediate right of the closing Options marker ('>>') and press <Alt-G>. This triggers the 'Document Assembly' routine. Of course, nothing is being assembled, but Pathagoras doesn't know that. It will still process any <<Options/Optional>> text blocks it finds.

Option 3: If you have created any DropDown Lists, you will see to the right of the list a toggle button marked either "Suppress" or "Process". See image above. (In Word2007/2010, this button appears to the left of the DDL trays.) Click it (and

depending upon its initial value, you may have to click it twice) and the text will be processed.

4.16.13 Negative Optional

'Optional text' is described elsewhere as 'keep it' or 'delete it' only. If the answer to the keep this text is 'Yes', the entire block is preserved. If 'No', the block is deleted.

That's not quite the whole story. Optional can accommodate a Yes/No or True/False reaction when a /NEGOPT (for negative optional) flag is inserted.

To enable the negative optional, just add '/NEGOPT' and some 'alternative' text. If value of the optional block is True, the text before the /NEGOPT is kept; if false, the text after the /NEGOPT is kept.

```
<<*Optional*!Apples!*An apple a day keeps the doctor away./NEGOPTThose
wormy apples didn't keep my doctor away.>>
<<*Optional*!Bananas!*My doctors tells me that bananas are the perfect fruit./
NEGOPTYes, I have no bananas.>>
<<*Optional*!Cherries!*Please top my [sundae/milkshake] with a cherry./
NEGOPTI don't want any cherries. They have pits.>>
```

It works with 'simple options' as well:

```
{!Apples!An apple a day keeps the doctor away./NEGOPTThose wormy apples
didn't keep my doctor away.}
{!Bananas!*My doctors tells me that bananas are the perfect fruit./NEGOPTYes, I
have no bananas.}
{!Cherries!Please top my [sundae/milkshake] with a cherry./NEGOPTI don't want
any cherries. They have pits.}
```

And it works with combinations of Simple and Robust optional blocks, and it also works if the !group! is set via an Interview.

4.16.14 Testing & Editing

Pathagoras provides an easy way to test, edit, retest, re-edit your <<*Options/Optional*>> text block structure (and any other text within <<double angle brackets>> without having to assemble a complete project.

The routine is called 'Process'. With the click of the **Process** button, Pathagoras will act upon the current document as if you were the end user in the middle of a 'document assembly' routine



Before getting started with 'Process', *save your work!* (If you process your work without saving it, you may be disappointed when your efforts are 'processed away.')

- To **process** the <<*Options/Optional*>>, {Simple Options}, <<*Repeat*>> and <<document>> blocks in your document:

Option 1: Click the Pathagoras Features menu. Click the 'Process' element from the list. (This is probably the 'cleanest' way, and the way we had in mind when we titled this section. But the next two options work equally well.)

Option 2: Place the cursor to the immediate right of the closing block marker ('>>') or "}" and press <Alt-G>. This triggers the 'Document Assembly' routine. Of course, nothing is being assembled, but Pathagoras doesn't know that. However, it 'thinks' it is in a document assembly routine and will process any <<block>> or {block} text it finds.

Option 3: If you have created any DropDown Lists, you should see a toggle button next to the List titled either "Suppress" or "Process". Click it (and depending upon its initial value, you may have to click it twice) and the on-screen text will be processed.

- **Structure and Integrity Checkers:** Pathagoras can review a source document (whether final, or simply 'under construction') to see if you are complying with 'the rules.' In many cases, it can automatically (with your permission) fix any errors it finds. So even before 'Processing' the document to see if everything works, run the 'Structure Checker' first and study its results.

To activate the **Structure Checker**, click "Wizards & Assistants | Structure Checker" item from the Pathagoras drop down features menu. (Structure Checker is also a pre-assigned Alt-Q | My Buttons element.)



If you have a large document, you may wish to test only a portion of it. To do so, simply copy and paste the portion you want to test to a new page. 'Process' just that section. Once the selected section is perfected, then test on the original, complete document.



If you are doing a lot of testing, you can elevate the "Process" and/or "Editing Tools" buttons to your [Quick Access Toolbar](#)⁶¹⁰.



Because all text within <<*Options/Optional*>> text blocks is 'plain text, you do not have to have Pathagoras loaded on your system in order to *create* them. You can create and fully edit them anywhere. This allows 'at home' and third party editing. Test them after you get them onto a 'Pathagoras' enabled computer.



This is not a clause 'testing' tip, but rather a document assembly testing tip. As you are testing how your documents are coming together, you may wish to *suppress* the processing of <<Options/Optional>> text block. You can tell Pathagoras to 'hold off' this processing by clicking the Suppress Processing box found in Utilities/Settings (look for the 'Document Assembly' tab in the 'Settings at a Glance' section).

Despite the above warnings to the contrary, do not worry *too* much about losing your work while testing. As you perform tasks in Pathagoras, remember that you are always in Word. You can click the Undo button (or press Ctrl-Z) and, in almost all cases, restore your work to its pre-processed state.

Pathagoras has a wide range of other tools to help you to identify problem spots. Check [this link](#)⁴⁴⁶ for how to access these editing tools.

See also:

[Structure/Integrity Checker](#) ¹⁷³

['The' Process](#) ¹⁷ (what we mean by 'processing' a document)

4.16.15 Administrative Text

When you construct <<*Options*/*Optional* and *Repeat*>> blocks, you should be aware of the placement of the asterisks, i.e., "*" or 'star'.

With properly placed asterisks, the text will not properly be read and processed by the program.

You will have at least two asterisks in any <<*Options*/*Optional* and *Repeat*>> blocks, maybe 3. All text before the closing asterisk is call 'administrative' text. This administrative text is not part of any final text that might remain in your document. It is, however critical to the smooth operation of Pathagoras.

Administrative text is at a minimum the introductory title to the text block. NOTE: The block titles *Options*, *Optional* and *Repeat* must always be surrounded by asterisks.

So, in this example,

```
<<*Options*Federal Express/USPS (First Class)/UPS>>
```

the administrative text is simply the title "<<*Options*". Here only 2 asterisks are used, and the final ">>" to close the block.

If, as in this longer example, you add short questions:

```
<<*Options*FedEx/USPS (First Class)/UPS*Per your request, the widgets will
be shipped by Federal Express and we will bill you for the extra cost of
shipping./Per your request, we will send the widgets by standard First Class
mail./Per your request, we ship your purchase via United Parcel Service.>>
```

the administrative text runs to the third (closing) asterisk: <<*Options*FedEx/USPS (First Class)/UPS*

Administrative text might also include a !groupname! you assign to a block to tie together <<*Options/Optional and Repeat*>> blocks that appear in disparate sections of your document.

No matter the use, the administrative text will always precede the 'final' (either second or third) asterisk in the block

4.16.16 Select Optional text via Options

Selecting *Optional* text using the actual value of an *Options* selection.

When you select a value from an <<*Options*!Groupname!*>> group, Pathagoras typically selects the appropriate value in a subsequent options block bearing the same !groupname! by using its *position* in the list. So, if you select the third choice of the <<*Options*!groupname!* . . .>> block, the same '3rd place' selection would be made for subsequent options blocks further down in the document.

But Pathagoras records both position of your choice and the actual ('hard') value of the choice. By doing so, **Options** blocks can feed an **Optional** text block using this 'hard' (as opposed to 'positional') value. If the 'hard' value being test for in the **Optional** text equation was selected in the initial **Options** choice, the **Optional** text will remain. Otherwise (as is typical of all Optional text), it will be deleted.

The formula to test for the 'hard' value in the subsequent Optional text block is simply :

```
<<*Optional*!Groupname = Value! . . . >>
```

Note: a {simple optional} formula will work just as well. The above could be constructed:

```
{!Groupname = Value! . . . }
```

If the 'hard' value of groupname is the same as the value selected from the original list, then the Optional text will be retained. Otherwise it will be deleted.

Multiple values can be provided, with each element separate from the others with a slash.

Further explanation and examples:

Positional ('classic') Method:

Using a classic Options block structure, you allow the user to choose which of 3 states the document is being prepared in.

```
<<*Options*!state!*Colorado/Virginia/New York>>
```

```
<<*Options*!state!*I love the mountains of Colorado./Virginia is for lovers./There is no place like New York!>>
```

When you select from the choices Colorado, Virginia or New York called for in the first Options line, Pathagoras records the position of the answer and returns the same positional value for values listed in the second Options line.

But sometimes the values in the initial options line (where the controlling selection is made) are numerous, but the need for responsive text applies to only a small set (perhaps just one item) of selection. E.g., if the selection is 'New York', insert the text, otherwise, leave blank. If only a positional evaluation were made, the second options list would have to contain 50 elements, all but the New York position (alphabetically #31 of the 50 United States) being blank.

To avoid that circumstance, you can create an <<*Optional*!groupname! . . . >> text block in which you evaluate the 'hard' value of the groupname as previously assigned. If the evaluation is true, the Optional text is preserved. If not true, it is deleted.

'Hard Value' Choice:

```
<<*Options*!state!*Alabama/Alaska/California/Colorado/Virginia/New York>> (Same as above, but with more choices.)
```

(in body):

```
<<*Optional*!state = Colorado!*Colorado has great skiing.>>
```

```
<<*Optional*!state = Virginia!*History buffs will find Virginia an incredible place to visit.>>
```

So if 'Colorado' is initially selected when the choices are presented from the 'top' line, the 'Colorado' language in the body would stay. The 'Virginia' language would be deleted.

If, on the other hand, 'Virginia' is initially selected, the 'Colorado' text would be deleted.

If any other state is selected, both optional blocks would be deleted.

Note: As you are setting up the text, remember that the evaluation that lead to action of an "<<*Optional* text" block is essentially 'true' or 'false'. If 'true', the optional block is preserved. If 'false', it is deleted.) By added the '=' to the !groupname!, you initiate that evaluation process on values that were initially not 'true/false'. ('Colorado' is neither 'true' nor 'false', but 'State=Colorado' can be.)

(You can cut and paste above examples into a document and see the action via <Alt-P>.)

You can also use '{simple options}' to set up the document;

```
{!state!Alabama/Alaska/California/Colorado/Virginia/New York>>
```

(in body):

```
{!state = Colorado!*Colorado has great skiing.}
```

```
{!state = Virginia!*History buffs find Virginia an incredible place to visit.} (in body)
```

You are not limited to a manually typed list. You can use MultiChoice Lists to supply the choices.

```
<<*Options*!state!**States*>> (at or near top; the double **** is correct.)
```

(States is a MultiChoice variable that ships with Pathagoras. Therefore, you can use these examples with no additional setup.)

```
<<Optional*!state = Colorado!*Colorado has great skiing.>>
```

```
<<*Optional*!state = Virginia!*History buffs find Virginia an incredible place to visit.>>
```

Remember: This feature is best used where it is impractical to code all 'subsequent' values in a long list. This should be particularly useful when a lot of choices must be presented in the initial request for value (perhaps, as in the above examples, all 50 United States are displayed in the selection screens), but only a few of those choices require extra text. (With the positional method, you would have to have a slash as a simple holding place for each 'no-extra text' situation.)

One more example. Let's say that California, but no other state, has a statutory requirement that certain text must be included in certain documents.

```
{!State!*States*} (at or near top)
```

```
{!State = California!This is a special clause required for California transactions only.} (in body)
```

4.17 {Simple Optional} Text Blocks

Pathagoras provides a very easy approach to providing optional text blocks. Almost no setup is required. Simply surround text you want to be 'optional' with {curly braces}.

Using this simple markup style is not as 'robust' as the <<*Optional*>> text blocks (for example, you cannot provide a customized question). But if the 'question' is inherent in the text itself, and the robust aspects of the alternative style are not needed, 'simple optional' is an ideal solution.

As stated above, to create a {simple optional text block}, place the boundary markers around the subject text. Unless you changed the default characters, the boundary markers are "{" and "}".

Like its 'big brother,' {Simple Optional Text} offers *two* types of blocks:

1. **“Optional” text:** This is pure ‘optional text.’ At document assembly time, the program will pause and ask “Include this text?” The user need only select “Yes” or “No” to tell Pathagoras whether the text block should be retained in the final document.

```
{The widgets you have ordered are not currently in stock. We will ship them as soon as possible. If we have not shipped within 5 days of this date, you will have the option to cancel the order.}
```

2. **“Options” text:** This ‘type’ allows the user to choose among several options. Each option is separated from the others by a simple, “/OR” (No quotes. Caps mandatory.). At document assembly time, the options are presented to the user on a selection screen. The user selects one or more of the options shown.

```
{As per your request, the widgets will be shipped by Federal Express. We will bill you for the extra cost of shipping./OR As per your request, we will send the widgets by standard ground transport. This may take 3 to 5 additional days./OR As per your request, we will hold the widgets for pickup by your courier./OR The widgets you have ordered are not currently in stock. We will ship them as soon as possible. If we have not shipped within 5 days of this date, you will have the option to cancel the order.}
```

i The **"/OR"** separator: The '/OR' separator will present the end user with mutually exclusive 'radio' buttons on the choice selection screen.

i The **"/ANDOR"** separator: If you want the end user to be able to select more than one of the options, separate the choices with **"/ANDOR"**. (No quotes. Caps Mandatory.)

When Pathagoras encounters the optional text block (#1 above), it will highlight the text in the document and ask if you want to keep it (Figure 1).

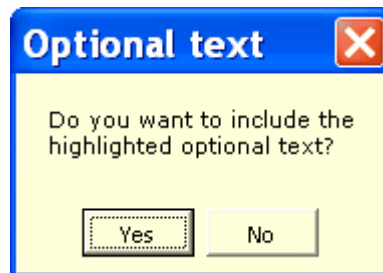


Figure 1. Optional Text dialog.
If you select <Yes>, the boundary markers are removed and the text remains in the document.
If you choose <No>, the entire text block is deleted from the document.

When Pathagoras encounters the options text block (#2 above), it will parse out the individual choices and display them onto buttons on a selection screen. (If the text is too long to fit, only the first 200 or so characters of the particular option will display.) Checkboxes are shown at the left of each choice so that you can choose more than one option, if desired.

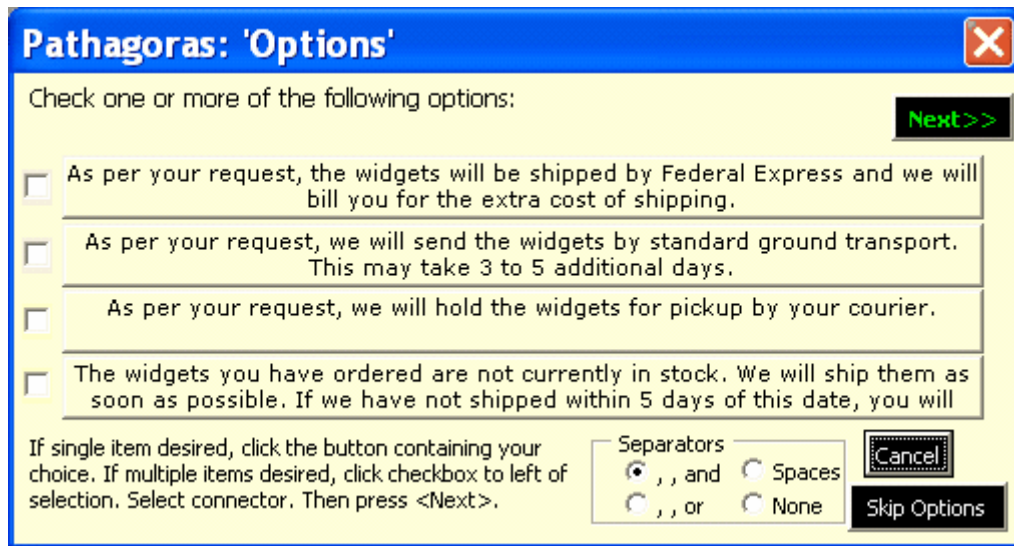


Figure 2. Options block dialog.

Note that the actual text of the particular option is provided (subject to space constraints.)

Note that the "/ANDOR" separator was used. It lead to a screen where the end user can choose one or more of the options.

!GroupNames!: Like its big brother, you can insert a !Groupname! to the beginning of related blocks of optional text. This will cause each {Simple Optional} and {Simple Options} containing the same !Groupname! to behave in a similar fashion.

Examples:

With Simple Optional text:


{!Disclaimer!We disclaim all liability if you try to shave your head with our chainsaw.} (blah blah blah . . .) and furthermore {!Disclaimer!we also disclaim responsibility if you try to use our hedge trimmer to trim your fingernails.}

With Simple Options text:


We suggest that the color for the living room be {!colors!red/ORyellow/ORburnt orange} and the contrasting trim color be {!colors!yellow/ORpurple/ORroyal blue}.


Note: The automatic selection of subsequent choices by Pathagoras when using !GroupNames! depends upon the *position* of the first choice in the list, not its answer. So, in the first above example, if 'yellow' (the second option) is selected when the first block is encountered, 'purple' (the second option in the second block) will be automatically selected.

Note: The same examples above appear in the section discussing [!Groupnames! with robust <<Optional/Options>> blocks](#)¹⁵¹. The differences are (1) the absence of the ability to add a short question or prompt for the end user, and (2) the requirement that '/OR' (instead of just a 'slash') be used to separate choices.

 **Note:** The {Simple Optional Text} module will identify and act upon any curly brace sets in your document. If a document already contains text within {curly braces} that you do not intend to be optional, you have 3 choices:

1. **Disable** Pathagoras' ability to process text it finds within curly braces. Click here to learn how to [enable and disable](#) ¹⁶⁹ this function.
2. Change the boundary markers that denote the 'native' text to something other than '{' and '}'. (Plain parenthesis will typically serve the same 'highlighting' function.)
3. Change the boundary markers that denote the 'simple' options text to something other than '{' and '}'. [Read more here](#) ¹⁶⁴.
4. Just forego simple options and use '[robust options](#)' ¹⁴¹.

 A {Simple Optional} text block cannot span more than a single cell of a table. {Simple Optional} text block can encompass an entire table, but the text block markers must be outside the table.

 **NESTING RULES:** You cannot nest another simple options block within within a simple options block. You cannot nest a robust <<*Options*. . .>> block within a simple options block. You cannot nest a <<*Repeat*. . .>> block within a simple options block. You **can** nest a robust <<*Optional*. . .>> block within a simple options block. You **can** nest plain variables within a simple options block, but not multiple choice variables. A 'best practice' would be to use simple options for their original 'simple' purpose, and don't try to nest anything (other than simple variables) within them.

4.17.1 Differences

While similar in function, the 'simplified approach' is more limited in function as compared to its 'big brother':

- the more powerful <<*Options/Optional*>> text blocks allows you to provide meaningful questions to the end user. See [Optional Text \(advanced\)](#) ¹⁴⁴.
- the more powerful <<*Options/Optional*>> text blocks allows you to provide a series of (typically shorter) 'questions' to reflect the options, instead of reproducing the actual text of the options; and
- the more powerful <<*Options/Optional*>> allow you to create sophisticated [Ask Tables](#) ⁴⁹⁹ which present a top of document listing of all <<*Options/Optional*>> questions found in the body of the document.

Because of the way that Pathagoras 'reads' the text in order to process it, Word 'fields' such as cross-references (which frequently contain "{" and "}" characters) can cause Pathagoras to act unpredictably. If you get such result, 'disable' the simplified {Optional} text block feature. (Do so via the main Utilities/Settings screen. The check box is at 'screen center.') Test again and see if the results improve. If they do not, contact Pathagoras Customer Service.

For many users, however, the simplicity offered by the {optional text} technique will surpass the benefits of the <<*Optional/Options*>> blocks. Please note that this feature may be used *in addition to*, and concurrently with, the <<*Options*/*Optional*>> Text blocks

discussed in the previous screen.

4.17.2 Converting

{ Simple Optional/Options } blocks are, well, simple. They are simple to create and simple to understand.

Here's a quick review on how to create { Simple Optional/Options } blocks:

- To create a { simple optional } block of text, surround that text with curly brackets. The text can be of any length.
- To create a { simple options } block of text, surround that text with curly brackets. If the choices are mutually exclusive, insert '/OR' (caps required) between each choice. If you want to be able to select one or more choices, insert '/ANDOR' (caps required) between each choice.
- To 'group' simple blocks (so the first selection of a simple options/optional block will automatically determine the selection for the remaining ones), insert a !groupname! (any text, any length up to 20 characters) immediately inside the opening curly bracket.

Remember: "Optional" means the entire block of text is keep or delete.

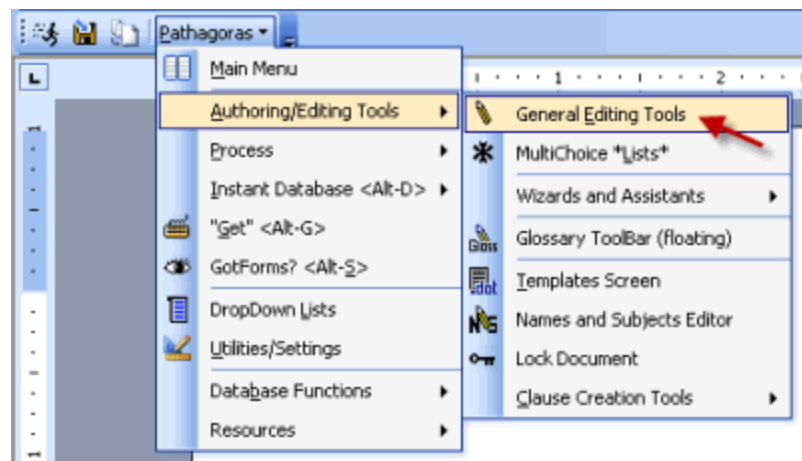
"Options" will list choices, one or more of which you intend to keep.

But with this simplicity come some limitations. They just don't have the robustness and flexibility that the more 'elaborate' <<*Options/Optional*>> blocks possess. Examples: You cannot attach questions to { Simple } text blocks. You cannot nest { Simple } blocks with each other to obtain a cascading answer pattern.

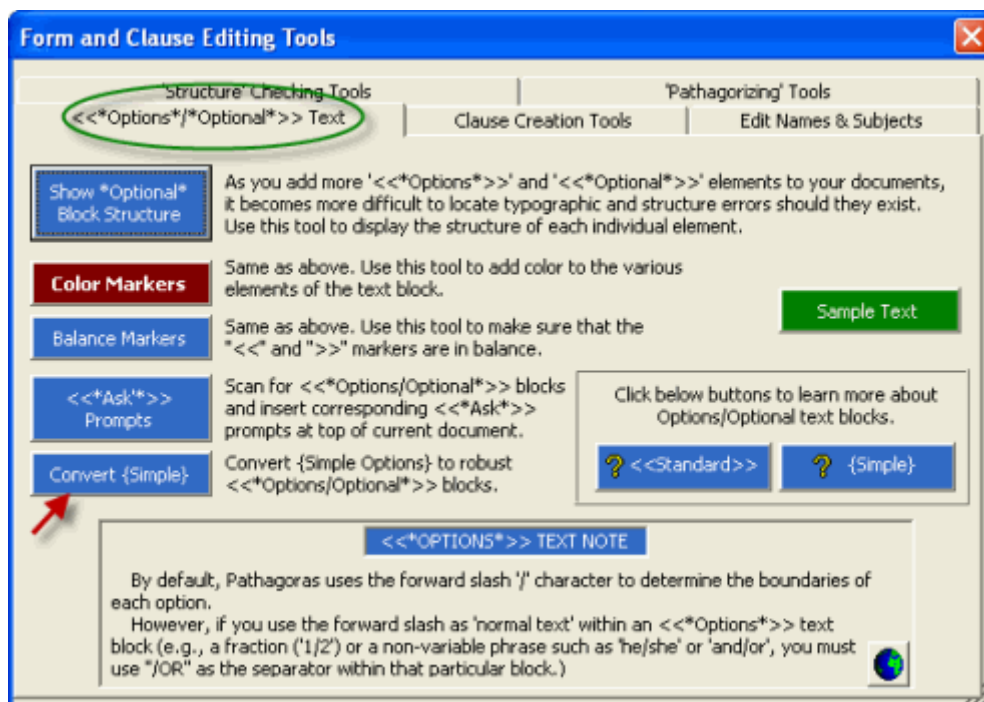
There likely will come a time that you wish to convert at least some your { Simple Optional/Options } text blocks to <<*Options/Optional*>> blocks. This will be especially so if you want to take advantage of Pathagoras' <<*AskOptions*>> table and more immediately, the <<*AskOptions*>> wizard that Pathagoras provides to automatically create those tables.

It is easy to convert { Simple Options } to robust <<*Options/Optional*>> blocks:

1. From Pathagoras' dropdown features menu, click "Editing/Clause Creation Tools" and then "General Editing Tools".



2. Select the <<*Options/Optional*>> tab from the screen that next appears.



3. Click the **Convert {Simple}** button.
4. Pathagoras will quickly identify and then convert any {simple options} blocks into the appropriate <<*Options*. . .>> or <<*Optional*. . .>> block. If no !groupname! has been assigned to the {simple optional} block, Pathagoras will ask for one as it converts each block. (If the {Simple} block already contains a !groupname!, it will be carried forward.)

See Also:

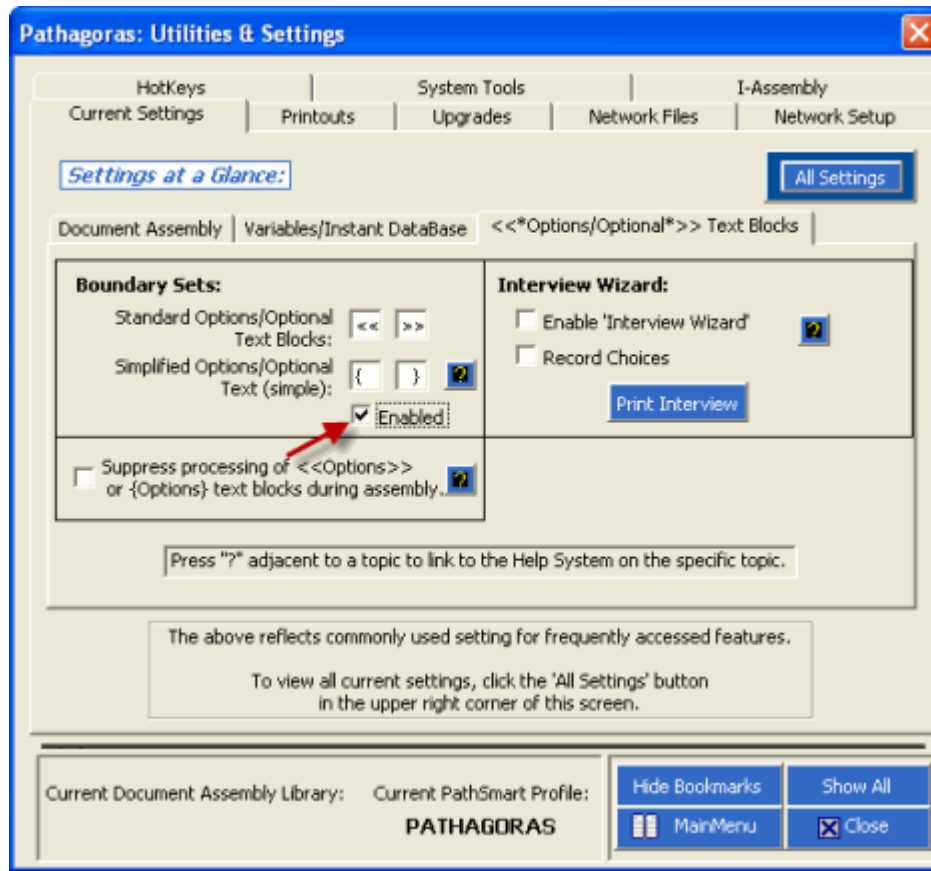
[!GroupNames!](#) 

4.17.3 Enabling/Disabling

Many users already have documents containing curly braces surrounding blocks of text. More likely than not these preexisting braces do not represent optional text. (Perhaps they have been used to mark out instructional or informational material in legacy documents.) If Pathagoras 'processed' {text} without warning, it would generate quite a bit of user confusion. {Simple Optional} ships in the 'enabled' state, but you can (1) disable the function or (2) you can change the character set that identifies a 'Simple Options' block to something else.

1. To enable/disable the module, call up the Utilities/Settings screen. Click the <<*Options/Optional* Text Blocks>> tab under 'Settings at a Glance' section. Next to the 'Simplified Options' label, change the characters or uncheck the "Enabled" box.

2. To change the boundary characters to another character set, type one or two 'mirroring' character sets in the appropriate text box. '{{' and '}}' will work, as will '<*' and '*>'. (When you type an opening boundary set, Pathagoras will automatically mirror the closing set for you.)



Of course, if you have a existing text already marked with { curly braces }, you might consider changing the boundaries of that text. Plain (parentheses) typically can serve the same 'highlighting' function, and they won't interfere with Pathagoras' operations.

4.17.4 Options (Mixing Simple with Complex)

If you use [!groups!](#)¹¹⁹ to link selections, you will be pleased know that you can pair {simple options} with the more complex sister using the !groupname! feature. That way, you can ask a meaningful 'question' at the top of the document but carry the answer forward to the remainder of the document using the less complicated {simple options} construct.

Example: The groupname !sex! is typically not descriptive enough to stand alone. An alternative would be to name the group "!sex of our client!" but its length might be cumbersome depending upon how many members of the group there are further in the document. So by using the robust AskOptions construct at the top of the document, we can ask a meaningful question, but use the simple grouname !sex! in the remainder of the document body.

```
<<*AskOptions*!Sex!Our client is a male/Our client is a female*>>

DEED OF CONVEYANCE

This Deed, by and between [Client Name], an unmarried
<<*Options*!sex!*male/female*>>, Party of the First Part, and . . .
```

```
The Party of the First Part conveys all rights and title that {!sex!he/
ORshe}has in the property, to the Party of the Second Part and further conveys
any inchoate or residual interests held by {!sex!him/ORher}in said
property. . .
```

4.17.5 Negative Simple Optional Text

The '**Negative Optional**' value: The default action of an 'no' or 'false' choice for {simple optional} text is to delete it.

However, when you tie several {simple optional blocks} together using !groupnames!, there occasionally is a need to delete the 'yes' choice but leave other text in the document that reflects the 'no' or 'false' choice.

Pathagoras lets you provide a 'false' value to simple optional text. This text will remain if you choose "No" or "False" to the question presented. To provide this 'false' part, simple type "/NEGOPT" (no quotes) at the end of the {simple optional text block} and then type the language you want to remain in the document if "No" or "False" is chosen by the user. (So, if the user answers True or Yes, the part before the slash remains and the false part is deleted, If the user answers 'False' or 'No', the part after the slash remains and the 'true' part is deleted.

Here is an example. Let's assume you are drafting a Will and want to provide for Guardianship of minor children (if any). You might first ask if there are any children, and if there are, name a guardian as their custodian if any are minors. If there are no children, you want to state that fact. Here is how that setup might look.

```
{!children!There were [number of children] born of the marriage.}
{!children!If any children who are still under the age of 18 years survive me, I appoint
[Name of Guardian] to serve as their guardian during their minority./NEGOPTThere
were no children born of the marriage.}
```

4.18 Dynamic Creation of Variables

Combining the techniques and principals set out in the above sections allow you some pretty powerful tools.

- **Dynamic creation of variables**

Let's say that you need two 'identical' documents for two people whose information is stored in a single data record. (A classic example of this would be reciprocal Wills one for the Husband and a substantially identical one for the Wife.) You could create two separate Wills (one for each) and use "[Testator]" and "[Spouse]" as the variables to identify the necessary parties templates.

But let's now say that you want to create both wills at the same time. By using <<*Options*>> text block in conjunction with a !GroupName!, you can cause Pathagoras to dynamically select the variables. Here is an example: <<*Options*!Whose Will!*[Husband Name]/[Wife Name]>>

With this setup, you could create both Wills in the same document. (To insert the proper 'other spouse' variables, simply reverse the listing of the variables, thusly,

<<*Options*!Whose Will!*[Wife Name]/[Husband Name]>>.

The table below presents a 'fuller' example of the above. Note that upon the selection of the first Option (which contains the !Who's Will! group, all variables are properly selected and ready for completion. This example can be copied from here and pasted into a Word document. Then, 'process' the document from the screen to cause Pathagoras to act on the <<*Options*>> blocks. The resulting document will be a 'perfect' reciprocal Will set, with all variables in their proper places.

Last Will and Testament of

<<*Options*!Whose Will!*[HUSBAND NAME]/[WIFE NAME]>>

I, <<*Options*!Whose Will!*[Husband Name]/[Wife Name]>>, being of sound mind, make this document my Last Will and Testament.

1. I give all of my property and estate to my <<*Options*!Whose Will!*wife/husband>>, <<*Options*!Whose Will!*[Wife Name]/[Husband Name]>>

2. I appoint my <<*Options*!Whose Will!*wife/husband>>, <<*Options*!Whose Will!*[Wife Name]/[Husband Name]>> to be my Personal Representative.

<<*Options*!Whose

Will!*[Husband Name]/[Wife Name]>>

=====Page Break=====

Last Will and Testament of

<<*Options*!Whose Will!*[WIFE NAME]/[HUSBAND NAME]>>

I, <<*Options*!Whose Will!*[Wife Name]/[Husband Name]>>, being of sound mind, make this document my Last Will and Testament.

1. I give all of my property and estate to my <<*Options*!Whose Will!*husband/wife>>, <<*Options*!Whose Will!*[Husband Name]/[Wife Name]>>.

2. I appoint my <<*Options*!Whose Will!*husband/wife>>, <<*Options*!Whose Will!*[Husband Name]/[Wife Name]>> to be my Personal Representative.

<<*Options*!Whose

Will!*[Wife Name]/[Husband Name]>>

- **Dynamic creation of 'identical' variables.**

Pathagoras offers another type of dynamic creation of variables in its <<Repeat>>

function. That function allows you to create multiple instances of essentially the same variable, but each resulting variable is properly 'incremented' to take into account multiple members of the group. This is discussed more fully (and with examples) at [this link](#)³⁸².

- **Dynamic selection of documents**

Instead of using <<*Options*>> and !Groups! to dynamically select the proper [variable] for a particular document, <<*Options*>>!/Groups! can lead to the selection and automatic insertion of a clause (or an entire document). This ability of Pathagoras to insert the content of text in this fashion is discussed more fully (and with examples) at [this link](#)¹⁰⁴.

- **A "Big" Example**

To see a larger example of the above, with many more Pathagoras elements in play, [click here](#)⁵²⁷.

4.19 Structure/Integrity Checker

Pathagoras can review a source document (whether in final form, or simply 'under construction') and determine if the document complies with 'the rules' pertaining to the proper construction of variables and options/optional blocks. It can also determine if the variables in the document align with the variables in a specified 'mask' (intake form) that is associated with that document.

Activating the Structure/Integrity Checker:

1. From the Pathagoras dropdown features menu, click "Wizards and Assistants | Structure Checker"
- or
2. Click the "Authoring/Editing Tools" item from the Pathagoras dropdown features menu. Select "General Editing Tools" and the click <Check Document Structure> button.
- or
3. Press Alt-Q and select the pre-assigned "Structure Checker" button (assuming that it has not been reassigned.)

When the "Structure Checker" has been called, Pathagoras will search out each 'Pathagorized' element in the current document (i.e., [variables], <<*Options/Optional*>> blocks and {Simplified Options}) and perform the following tasks:


1. It will make sure that all boundary markers are in balance. If Pathagoras detects an imbalance, it will stop and highlight the most likely location where the imbalance exists. (The boundary markers checked are '[' and ']' (or other variables boundary markers); '<<' and '>>' (or other options/optional boundary markers); and '{' and '}' (or other simple options boundary markers).
2. It will check for proper structure of each of your <<*Options/Optional*>> blocks, including
 - the proper placement of asterisks around the key terms;

- the proper structure of any nested elements;
- the proper composition of any !group names! and 'prompt' (short question) text;
- the proper placement of the final asterisk closing the administrative section of the text block.

If Pathagoras detects a structural error, it will stop and highlight the most likely location of the error.

3. It will ask if you wish to compare the variables in the document with those of a specific mask (intake form) that you have earlier created and saved. If you answer yes, it will

- ask you to select the mask;
- scan the document for the existing variables and for each found, compare it to those in the mask; and
- stop at each variable that does not have a corresponding match in the mask so you can decide whether to amend the variable, keep it and/or add it to the mask.

 When you initiate 'Structure Checker,' Pathagoras will ask if you want to perform a check on just the displayed document or all documents in the parent folder that contains the initial document. If you select the 'All Documents' option, Pathagoras will not return as many 'ok' messages or identify as many error points as it would if it were checking a single document. Rather, will simply identify those documents in the parent folder which need attention. After the complete routine has completed, you should return to the document identified as containing a structural error. Then, run the same routine in the individual document and repair as suggested.

See next page in the Manual for more tools to help you inspect the document for consistency with structure rules.

4.20 Debugging Variables & Options Text

Pathagoras provides several tools to help you to highlight and 'debug' problems with variables, <<*options*>> and {simplified options} text.

Alphabetize:

A slight misspelling between two instances of what is intended to be the same variable (or the same !group! name) will cause Pathagoras to process two separate entries. If the entries appear on two separate IDB pages, it may be difficult to detect the error. In order to see similarly spelled variables in closer proximity, you can tell Pathagoras to alphabetize the entries:

- Display the IDB screen. Scan the document (or recall the mask) you want to 'debug'.
- Press the <Utilities> button and then press the <Alphabetize Variables> button which will appear in the bottom portion of the screen.
- Note any naming errors. Return to the document itself and correct the spellings. The next time you scan for variables, the 'bug' should be gone. (You can even ask Pathagoras to take you directly to the first instance of the selected variable. The button for that purpose is immediately below <Alphabetize Variables>.

If the error is a mask error, please note the following: There is not a 'return to previous

state' button. This is not a problem if you wish to save the mask in its alphabetical order. Make the corrections and save the mask in its current state. If you don't want to preserve the alphabetical listing, make a note of the offending entry. Then reselect the mask so that its original display returns. Then navigate to the offending entry and correct it.

Paint the [brackets]:

Pathagoras can 'paint' the brackets so it is easier to distinguish them from the surrounding text. Click the Editing Tools entry in the Pathagoras drop down features menu. Click the <Color Markers> button from the Form and Clause Editing Tools screen that will appear. (See red arrow in Figure below.)

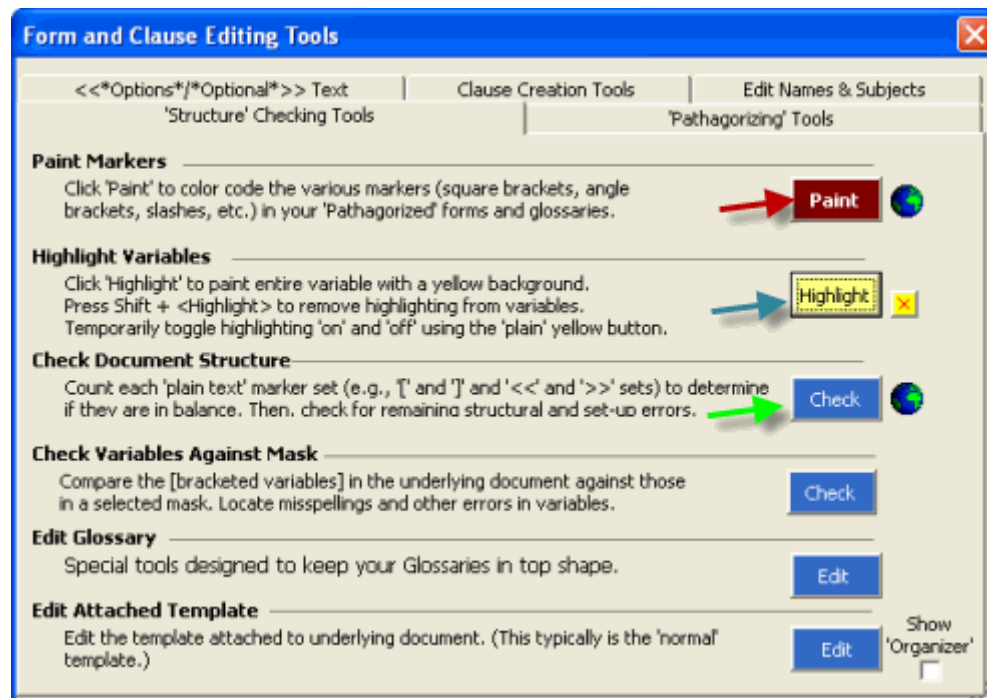
Highlight the [variable]:

To 'paint' the entire variable, check the box beneath the <Color Markers> button. (You can also click <Highlight each variable in yellow> which appears on the Instant Database screen when you click the <Utilities> button.) (See blue arrow in Figure below.)

➔Note: Highlighting a variable is actually the insertion of background 'highlighting.' The highlighting can be toggled on and off by clicking the yellow button just to the right of the checkbox. The highlighting can also be toggled on and off via Words Tools|Options|View menu: check, or uncheck, the Highlight box.

Count the brackets:

To determine if you have an equal number of brackets in the document, click the Editing Tools entry in the Pathagoras drop down features menu. Click the <Balance Markers> button from the Form and Clause Editing Tools screen that will appear. (To check just a section of the document, highlight that section before running this feature.) (See green arrow in Figure below.)



Painting or highlighting variables makes them easier to see and helps in the 'debugging' process.

4.21 Changing Variable Names

Sometimes the name you initially assign to a variable turns out not to be the 'best' choice.

Let's say you called a variable [Date] and then decided it should be [Date of Contract] so that it would be more meaningful to the end user.

Or, perhaps even more widespread, you assigned a variable [Name of Client] and days or weeks later decided that [ClientName] fit your naming scheme better (to make it consistent with [ClientAddress], [ClientCity, ST ZIP].


Or, perhaps you decided that you want to break up a variable like [ClientName] into its components, such as [ClientFN], [ClientMI] and [ClientLN].

Not to worry. You can easily change any variable to a value more to your liking at any time.

If you are just getting started, and have only created a few variables, manually changing them is probably the easiest course. But if you have spent a lot of time creating variables before you decided a different name(s) was in order, see [Search & Replace](#) ⁴⁴⁸ for suggestions on how to make those changes in a global fashion.

4.22 Anatomy of Variable and Optional Text Characters

Here is a collection of the various character sets for marking variables and optional text. They are listed from the simplest to the most complex:

 The boundary and other colorings that you will see in the text markers below is for emphasis only. No marker coloring is required.

The anatomical discussion of each example is set out in blue:

VARIABLES:

Simple variable: [Client Name];[quantity] **Anatomy:** Two facing square brackets (the *boundary* characters) surrounding a core term. The core term can one word or a short phrase or itself can have structural components.

Multiple choice variable: [chocolate/vanilla/strawberry] **Anatomy:** A simple list of optional terms within boundary characters. Each choice separated by a forward slash.

MultiChoice variable: [*States*]. **Anatomy:** The two asterisks indicate the call to a 'MultiChoice List' A MultiChoice List is a series of optional values (in this case the 50 United States) stored in a separate file and referenced by a simple 'alias.' The core term is the alias and it is surrounded by the boundary characters.

Titled variable: [Special Order: Yes/No/Not Applicable] . . . [Priority Mail: Yes/No/Not Applicable] **Anatomy:** A title can be added to assist the end user in completing the Instant Database screen. Regular variable structure. The 'Title' is typed immediately after opening boundary character and 'closed' by the colon. Titles can be used for

simple or multiple choice variables.

Grouped variables: [**!client!**he/her/they] . . . [**!client!**his/hers/theirs] **Anatomy:** A groupname is used to tie two or more variables together. (The positional selection of the first variable automatically causes the same positional selection of subsequent variables.) Regular variable structure. The 'groupname' is typed immediately after the opening bracket between '!' (exclamation marks). Groups are appropriate only for multiple choice variables.

OPTIONAL/OPTIONS TEXT BLOCKS (simple)

Simple Optional Text (Keep or delete the whole block.)

{ Shipping included in cost } **Anatomy:** Two facing curly braces (the *boundary* characters) surrounding a core term. The core term can range from a single character to multiple paragraphs.

Simple Optional Text with !Group! name (A !group! name is used to tie two blocks together. The selection or rejection of one member of the group results in the same action for the remaining members of the group.)

{ !shipinfo!Shipping included in cost. } . . . { !shipinfo!Remember shipping charges have already been included in the list cost of the product. } **Anatomy:** Two facing curly braces (the *boundary* characters) surrounding a core term. The 'groupname' is typed immediately after the opening brace, between '!' (exclamation marks). The core term can range from a single character to multiple paragraphs.

Simple Options text (Select one or more elements from block.)

{ Shipping included in cost/ORShipping Costs Extra } **Anatomy:** A simple list of optional terms surrounded by the boundary characters. Each option separated by the characters "/OR" (no quotes, but the capital 'OR' is mandatory). To allow multiple selections, use /ANDOR as separator.

Simple Options text with reference to MultiChoice *List*.

{ *States* } **Anatomy:** The two asterisks indicate the call to a 'MultiChoice List'. A MultiChoice List is a series of optional values (in this case the 50 United States) stored in a separate file and referenced by a simple 'alias.' The core term is the alias and it is surrounded by the boundary characters.

Introduction: OPTIONAL/OPTIONS TEXT BLOCKS ('robust')

The setup for Pathagoras' more robust <<*Options*/*Optional*>> text blocks shown below is not without its complexities. Compared with the setup required by competitive programs, however, it is undeniably simpler. Because it is plain text, you don't have to fish around for the right field coding and switches. Seen 'on screen' when the source text is recalled, it actually becomes quite 'readable' once you become familiar with the 'anatomy.'

The precise placement of the various characters that make up the blocks is mandatory. Not to worry too much, however. Pathagoras provides tools (1) to automate the creation of <<*Options*/*Optional*>> blocks and (2) to check the structure of each block after they have been created.

The '**call to action**' is the key term in the text block that tells Pathagoras what to 'expect' from the remaining sections of the text block. It will be either Optional, Options or Options(radio).

The '**administrative section**' is that part of the text block that contains the 'call to action' and the

group and display information. It does not remain in the final assembled document.

OPTIONAL TEXT BLOCKS (robust)

Basic:

<<*Optional*You may also pick up your block of cement from our help desk between the hours of 9 a.m. and 5 p.m.>> **Anatomy:** Two sets of 'facing' angle brackets (the **boundary** characters) surround the core block. The red is only for emphasis. The word "Optional" surrounded by asterisks constitutes the 'call to action.' The actual optional text can range from a single character to multiple paragraphs. Don't forget the closing brackets at the end of the block.

With question:

<<*Optional*Is pickup available?*You may also pick up your block of cement from our help desk between the hours of 9 a.m. and 5 p.m.>> **Anatomy:** Boundary characters surround the core block. The word "Optional" surrounded by asterisks constitutes the 'call to action' followed immediately by the prompt or question that you want to be shown to the end user during document assembly. A third asterisk closes out the 'administrative section' of the block. The actual optional text can range from a single character to multiple paragraphs.

With !Group! (A !group! name is used to tie two blocks together. The selection or rejection of one member of the group results in the same action for the remaining members of the group.):

<<*Optional*!pickup!*You may also pick up your block of cement from our help desk between the hours of 9 a.m. and 5 p.m.>> (body text . . .body text)

<<*Optional*!pickup!*Be sure bring a crane with you.>>

Anatomy: Boundary characters surround the core block. The word "Optional" surrounded by asterisks constitutes the 'call to action.' The 'groupname' is typed immediately after the call to action between '!' (exclamation marks). A third asterisk closes out the 'administrative section' of the block. The actual optional text can range from a single character to multiple paragraphs.

With !Groups! and Question/Prompt:

<<*Optional*!pickup!*Is pickup available?*You may also pick up your block of cement from our help desk between the hours of 9 a.m. and 5 p.m.>> (body text . . .body text)

<<*Optional*!pickup!*Be sure bring a crane with you.>>

Anatomy: Boundary characters) surround the core block. The word "Optional" surrounded by asterisks constitutes the 'call to action.' The '!group! name' is typed immediately after the call to action between '!' (exclamation marks) followed immediately by the prompt or question that you want to be shown to the end user during document assembly. A third asterisk closes out the 'administrative section' of the block. The actual optional text can range from a single character to multiple paragraphs. (The prompt would appear only in the first instance of the group. It does not need to be repeated in subsequent members of the group.)

OPTIONS TEXT BLOCKS (robust)

Basic:

<<*Options*I have no children./I have one minor child./I have [num cdn] minor children.>>

Anatomy: Two sets of 'facing' angle brackets (the 'boundary' characters) surround the core block. The word "Options" surrounded by asterisks constitutes the 'call to action.' The various choices are separated by forward slashes. The actual text of the options will appear as the 'prompts' on the selection form during document assembly process.

With reference to MultiChoice *List*

<<*Options**States*.>> **Anatomy:** The 'boundary' characters followed by the call to a 'MultiChoice List' alias.

With Prompts/Questions (up to 6 choices and 6 responses)

<<*Options*No children/One child/Two or more children?* I have no children./I have one child to whom I give, devise and bequeath the remainder of my estate./I have [number of children] minor children to whom I give, devise and bequeath the remainder of my estate, in equal shares.>> **Anatomy:** Boundary characters surround the core block. The word "Options" surrounded by asterisks constitutes the 'call to action.' The optional term '(radio)' is included in this call to action to indicate that the choices are mutually exclusive. The call to action is followed immediately by a series of prompts or question that you want to be shown to the end-user during document assembly. A third asterisk closes out the 'administrative section' of the block. The options text follows, each option separated from the other by forward slashes. Each option can range from a single character to multiple paragraphs.

With !Group! (A !group! name is used to tie two blocks together. (The positional selection of the first variable automatically causes the same positional selection of subsequent variables.):

<<*Options*!Children!*I have no children./I have one minor child./I have [num cdn] children.>>

<<*Options*!Children!*I give the remainder of my estate to the following persons./I give the remainder of my estate to my child./I give the remainder of my estate in equal shares to my children, per stirpes.>>

Anatomy: Boundary characters surround the core block. The word "Options" surrounded by asterisks constitutes the 'call to action.' The '!group!' name is typed immediately after the call to action between '!' (exclamation marks). A third asterisk closes out the 'administrative section' of the block. The actual options text, each separated from the other with a forward slash, can range from a single character to multiple paragraphs.

With !Group! and Questions/Prompts:

<<*Options*!Children!No children/One child/Two or more children*I have no children./I have one minor child./I have [num cdn] minor children.>>

<<*Options*!Children!*I give the remainder of my estate to the following persons./I give the remainder of my estate to my child./I give the remainder of my estate in equal shares to my children, per stirpes.>>

Anatomy: Boundary characters surround the core block. The word "Options" surrounded by asterisks constitutes the 'call to action.' The '!group!' name is typed immediately after the call to action between '!' (exclamation marks) followed immediately by the prompts or questions that you want to be shown to the end user during document

assembly. A third asterisk closes out the '*administrative section*' of the block. The actual options text, each separated from the other with a forward slash, can range from a single character to multiple paragraphs. (There must be as many prompts as there are actual choices. The prompts would appear only in the first instance of the group. They do not need to be repeated in subsequent members of the group.)



You can have only six `*Options*` in a {Simple Options} or robust `<<*Options*>>` block. There are practical reasons for this, not the least of which is the amount of screen 'real estate' that is needed to display the various options in the Options dialog. There are, however, two good 'work arounds' to this limit.

1. Break the Options block into two or sets of (up to) six choices. Create a 'super' options block encompassing the (nested) others. The 'super' level `<<*Options*>>` will be presented first. Selecting a choice from that level brings up the (smaller) sub-group.
2. If each of the options is relatively short (a few hundred characters or so), create a [MultiChoice *List*](#)⁴⁵⁸ of the various options. This routine is easy to implement and will solve most of the 'problems' caused by the limit of 'six.' The options (and there could be dozens of them in the MultiChoice `*List*`) are referenced in the source text in this remarkably compact fashion: `<<*Options**mylist*>>`
3. If you use prompts or questions, you need them only in the first instance of the group. The prompts/questions do not need to be repeated in subsequent members of the group.

4.23 Samples DropDown List

Pathagorizing a document is not hard, but it can be tedious. It takes time to go through all of your documents adding the variables and Options/Optional/Repeats blocks that help to automate the process of document assembly.

Pathagoras (like every other program on the market) requires a specific syntax when you want it to perform a specific task. Stray from the syntax and it won't know what you want it to do.

Memorizing all the commands is impossible -- there is just too much in the program to allow for that. Looking up the syntax in this Manual for a particular command (or even remembering that a particular command is available, for that matter) can slow you down.

We hope to make the document authoring process significantly easier by providing a DropDown List containing an organized collection of sample text. This list serves a two fold purpose:

- it provides the proper syntax for the command you are seeking
- reminders of the various commands that are available.

Because the Samples are in a DropDown List, it is always on and always available, and is just a point and a click away. Click on an entry to read more about the feature. Peruse the wide variety of examples that are available.

To add the Samples to your DropDown List collection, click 'Menu' to the right of the DropDown List section of the screen. At the bottom of the menu that will appear, click the bottom-most button called 'Samples' Collection. In just a few seconds, you should see a new DropDown List titled Samples Glossary. Drop down the list and click away.

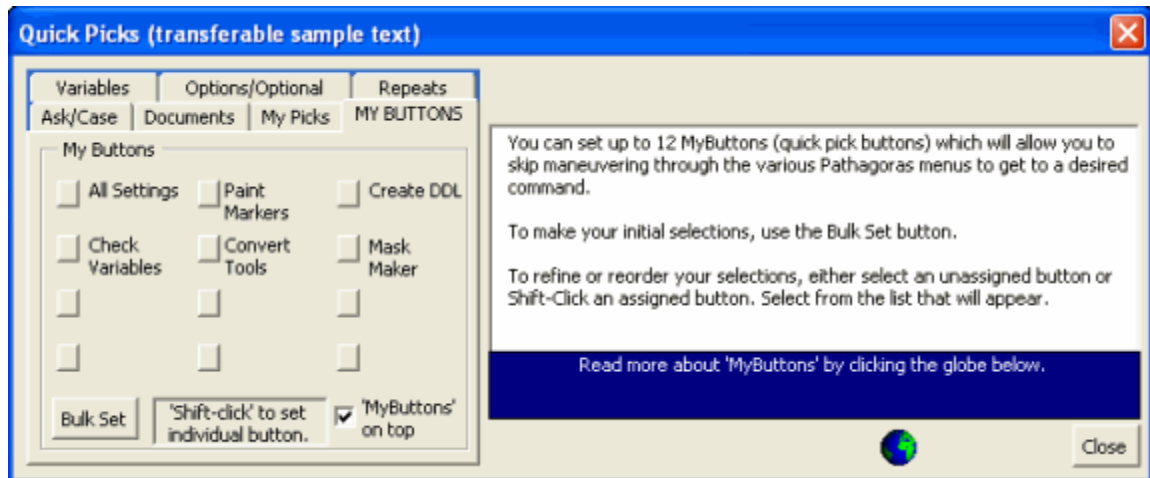
5 Quick Picks

Word (and other Office programs) offers the Quick Access Toolbar (QAT). You can place any command you can run by click a series of menu choices onto the QAT, thus avoiding the navigation and button clicking otherwise required.

You can add just about any Pathagoras command to your QAT. And we encourage you to do so. Read more about that [here](#).

But sometimes you want to keep your Pathagoras commands handy but in a single location (instead of with your other Word commands). And you can with QuickPicks

With Quick Picks, you can assign up to 15 Pathagoras functions to buttons. To activate Quick Picks, press Alt-Q. At the resulting screen, select a button and then a command. (Note that Pathagoras has pre-assigned a few popular commands. You can overwrite the assignments or keep them and add new ones to the remaining unassigned buttons. You can add 'in-bulk' using the button at the far left/bottom, or set them one at a time. You can even change a label (the text to the right of the button) if you don't like the label we created.



These are some of the choices you can place into your MyButtons collection:

Document Assembly	Initiate a document assembly session via Library & Books (same as clicking the Document Assembly button in the Pathagoras toolbar)
All Settings	Show All Settings Screen
Paint Markers	Paint Pathagorizing markers in vivid colors to make them more visible
Create DDL	Create a new DropDown List
PathSmart	Document management feature: Quickly access your favorite folders
SaveSmart	Document management: Quick save to favorite folders
Check Variables	Compare variables in underlying document to set list of variables in a mask

Structure Checker	Check Structure of document elements (variables, options blocks, etc.)
Process (test)	Test Options, Optional and Repeat blocks (Same as <Alt-P>)
Glossary Toolbar	Show the floating Glossary Toolbar
Web Browser	Show Web browser
Variables Wizard	Display 'Create Variables' Wizard
Options Wizard	Display 'Create Options' Wizard
Conversion Tools	Convert a folder to a glossary, a glossary to a folder, etc.
Convert HotDocs	Convert HotDocs fields to Pathagoras variables
MultiChoice Edit	Edit your MultiChoice *Lists* ('star' variables)
TermWorks	Show 'Term Works' screen
Toggle Bookmarks	Display/Hide bookmarks
Instant Database	Display Instant Database screen (Same as Alt-D)
IDB Settings	Instant Database Settings & Tools screen
External database settings	Display settings to external database links
Prefix Table	Show Prefix Table
Mail Merge	Perform Mail Merge on underlying document with database or Instant Database records.
Mask Maker	Scan document or all documents in folder to create intake form ('Mask')
Name & Subjects Editor	Display names of every document in selected folder in easy to edit listing. Quickly modify names and/or subjects to meet your document assembly needs (alphabetize, add sorting prefixes, etc.)

The list that ships with Pathagoras is augmented on a fairly regular basis, and your installed version of Pathagoras likely has more choices than those listed above.

If you have a 'favorite' button that you wish for us to add to our list so you can add it to your Quick Picks collection, [just ask](#).

The Pathagoras System

Instant Database Screen

Part



VI

6 Instant Database Screen

No document assembly system is complete without an efficient means by which it can identify the variables in the document and replace those variables with personal data.

The Instant Database module provide this feature.


Definition: 'Variables' are place holders that strategically placed in your document. After the document is fully assembled and processed, these variables stand ready to be replaced with personal data before the final document is signed, mailed or otherwise finalized.) See

[Variables](#)  117.

Discussion:

After you have 'assembled' a document by calling in the desired blocks of text, making your multiple choice decisions and deleting those text sections that are not relevant to your client's need, it likely will not yet be ready for publication. Chance are your source text contained bracketed [variables] that you will need to replace with *personal values*. Those [bracketed variables] need to be replaced with appropriate names, addresses, quantities, colors, etc. to reflect the objectives of the writer and the needs of the client/customer.

The primary variable replacement tool is the 'InstantDatabase' ('IDB') module. IDB is the topic that will be discussed in the sub-parts to this section. The IDB screen is displayed by pressing the keyboard hot-key combination <Alt-D>.

(A second variable replacement tool, called [GotForms?](#)  348 in a separate section to this manual. It is activated by pressing the hot-key combination <Alt-S>. It is a less used tool, but you should be familiar with it in case it fills a need.)

6.1 The basics

Pathagoras' Instant Database module uses a straightforward 'find & replace' scheme for replacing document variables with personal information. Indeed, the IDB system is nothing more than a fancy adaptation of Word's Find & Replace (Ctrl-H) feature. The differences are these:

- It works much more quickly than you could perform any single step in the process manually.
- It can act on up to 999 individual variables per session (30 to 60 is typical, but it is nice to know that there is no practical limit).
- It can handle multiple choice variables and other 'setups' that Find & Replace cannot.
- A variable can be automatically completed based on the answer to, or value of, another variable.
- It can store all of your answers in a reusable database. Find & Replace cannot do this.



"Instant Database" is so named because of its ability to 'instantly' create client, patient and customer records and to maintain a database of these records with little effort on your part.

Be aware, however, that at its core, Instant Database is simply a powerful "find & replace" tool. You can use Instant Database for *all* of your find and replace functions. It is not limited to replacing text in 'Pathagorized' documents.

To use Instant Database in this fashion, simply press <Alt-D> to display the screen. Then, type a value (a word or phrase that you want to replace) at the left of the IDB screen. Type a replacement value at the right. Repeat for as many items as you wish. Press **Next>>** and it's done! (If you want to save the values for use on other documents, feel free to do so.)

Instant Database is activated from the keyboard by pressing <Alt-D>. Here is a screen shot of what you will see.

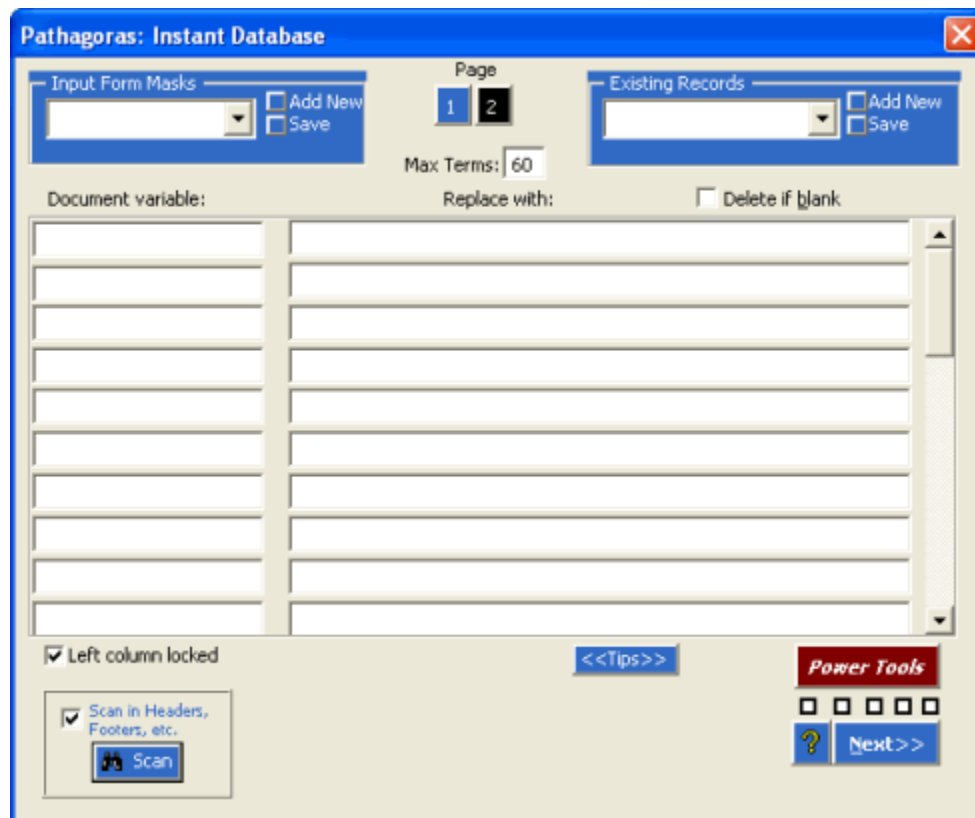


Figure 1. Instant Database Mask
(initial view)

- The left side of the screen will contain the document's "variables" Variables are 'holding places' strategically placed throughout the document which are waiting to be replaced with 'real' names, places, dates and other value. To initially populate this section, press the <Scan> button in the lower left quadrant of the screen. Pathagoras will quickly scan the underlying document for bracketed variables and place them in the left column.
- The right column is where the user will insert the 'replacement' values (personal data)

for each variable.

- When you are ready to signal Pathagoras to replace the variables at the left with the values typed at the right, click the [Next>>](#) button in the lower right corner of the screen.
- To access a series of helpful tools and utilities, click the red Power Tools button.
- The Input Form Masks dropdown contains all available 'masks' (pre-designed input templates, containing variables and completion tips)
- The Existing Records dropdown contains all of your saved data records, in alphabetical order. (The top of the list also displays the 9 most recently used records for easy selection.)



You can resize the Instant Database screen to your liking. If the 'Variable' or 'Replace With' fields are not wide enough, click the red [<Power Tools>](#) ⁴²² button. Look for the Resizing buttons to change the width or height of the screen.

See also:

[Variables](#) ¹¹⁷

[Multiple Choice Variables](#) ¹¹⁸

[Groupings](#) ¹¹⁹

[File Location](#) ²⁵⁰

[Power Tools](#) ⁴²²

[More Tools](#) ²¹²

6.2 Scan

To Scan a document for bracketed variables:

1. Press the <Alt-D> key combination to display the Instant Database screen. The screen will probably be blank. See **Figure 1**, below.
 - If the AutoScan feature has been selected, the screen will be populated with terms located in the AutoScan in the left column. (In other words, the next step was automatically performed.)
2. If AutoScan did not populate the screen, click the blue [Scan](#) button in the lower left portion of the IDB screen.
3. Pathagoras will display the results of the scan into the left column of the IDB screen. **Figure 2.** (If multiple choice variables were discovered during the scan, the choices are show in the right column, but otherwise the right column will be blank, awaiting replacement values from the user.)

➔Notes:

- All bracketed words are presumed to be the document's variables.

Occasionally, Pathagoras will identify a bracketed term that you did not intend to be a variable.

Example: Author and secretary initials are frequently presented at the foot of a letter in brackets. they will be picked up on the scan. When that happens, you can simply ignore it. It will remain in your document perfectly intact.

- Pathagoras' default 'enclosing' characters for Scanning are '[' and ']'. However, you can change the default characters to any character or character sets of your liking. They can be a single character (more traditional) or up to 3 characters (e.g., "[*&" and "&*]")

Exception 1 to the above 'anything goes' rule: The enclosing characters "<<" and ">>" are reserved by Pathagoras for other purposes. They cannot be successfully used as enclosing characters for the Instant Database system.

Exception 2 to the above 'anything goes' rule: The enclosing characters for variables must be different from those chosen for 'Simple Options' ¹⁶⁴. While it may seem obvious, it is still a 'rule'.

4. 'Scope' of the Scan. By default, Pathagoras will scan for variables in just the 'main' document (and not within headers, footers, textboxes and other non-main document locations). However, if your documents contain (or may contain) variables in headers, footers, etc., just check the box just above the Scan button that reads "Scan in Headers, Footers, etc." The setting is 'sticky' and will remain set for all future sessions (unless, of course, you uncheck it).
5. Once the scan is done, your job is to provide replacement data for each variable. Hand-type the personal data to replace the variables. **Figure 3.** It is perfectly fine to leave the value for a variable 'blank.' You can complete it later.
 - If a Data Record was previously saved for a specific client or customer, you can simply recall the Record. It will appear under the Existing Data list in the upper right section of the screen. No retyping required (unless of course, you need to change a specific value from the saved record).

Figure 1. A 'blank' Instant Database screen. The 'search for' text (typically a Document Variable) appears in the left column. You would provide the 'Replace With' text in the right column.

The screenshot shows the 'Pathagoras: Instant Database' window. At the top, there are tabs for 'Input Form Masks' and 'Matter Data'. Below these, there are checkboxes for 'Add New', 'Save', and 'Delete'. A 'Page' indicator shows '1' of '2'. A 'Max Terms' field is set to '60'. The main area is divided into two columns: 'Document variable:' and 'Replace With:'. The 'Document variable:' column contains a list of variables: [Testator], [Testator Address1], [Testator Address2], [Testator City], [Testator State], [Imarried!am/am not], [Spouse], [husband/wife], [Children], and [Guardian]. The 'Replace With:' column contains empty text boxes for each variable. A 'Delete if blank' checkbox is checked. At the bottom, there are buttons for 'Show Tips', 'Power Tools', and 'Next >>'. A 'Scan' button is also visible.

Figure 2 Document scanned. Bracketed variables placed in left column. Multiple choice variables are shown in dropdown lists.

The screenshot shows the 'Pathagoras: Instant Database' window with the same layout as Figure 2, but with data entered. The 'Document variable:' column contains the same list of variables. The 'Replace With:' column contains the following values: Peter Pathagoras, 123 Sparta Boulevard, Apartment 3, Athens, Georgia, am, Mary Q. Pathagoras, wife, Plato Pathagoras and Audrey Pathagoras, and Socrates B. and Helena S. Jones. The 'Delete if blank' checkbox is checked. At the bottom, there are buttons for 'Show Tips', 'Power Tools', and 'Next >>'. A 'Scan' button is also visible.

Figure 3. A 'completed' Instant Database form.


See Also:

[Data Records](#)  192









[Masks](#)  260

6.3 Completing a Variable

To complete a variable, just type in the right column the value you want the variable to represent. If the variable is multiple choice, select the entry from the drop down list presented. More on multiple choice list below.

You should provide your values in upper and lower (or otherwise 'normal') text. The emphasis, case, color and other attributes of the text in the source document control how the text ultimately will be formatted when it replaces the variable. The exception to this is if you type the answer in ALL CAPS, the variable, when replaced will be in ALL CAPS. See "[Emphasis](#)"  for more information.

There is no rule that requires your answer to be 'text.' Your answer can include a wide variety of references, shorthand marks, etc: Here are a few, with links to a fuller discussion:

- [<<FileName>>](#) : Insert an entire document in place of the variable, not just a few characters of text.
- [<<FolderName>>](#) : Select a file to insert from a designated folder.
- [<S>pell out number](#) : You type the number, Pathagoras 'spells it out'.
- [<\\$>Currency](#) : Convert number into spelled out currency.
- [Equivalency](#) : Make one variable in the IDB screen equivalent to another.
- [Dates](#) : Select, or calculate, date from a calendar.
- [Math Functions](#) : Do simple math using the values of other variables.
- [<P>aragraph](#) : You can insert a paragraph marker into your answer with this simple text addition
- [<URL:. . .>](#) You can insert a call to a website or any other location.
- [<Mailto: >](#) You can insert a Mail To command within a document



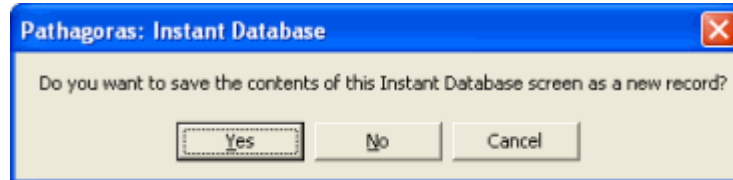
When you finish completing all your variables, there are 3 buttons you will consider clicking.

- The "Add New" button will create a new record based on the current on-screen data.
- The "Save" button will update the record showing in the Existing Records text box (assuming that you called up an existing, or earlier added a) record. (Pressing "Save" when there is no record showing in the Existing Records text box is the equivalent of "Add New."
- The "Next" button. *This is most often your best choice.* When pressed, Next will ask if you want to create a new record or (if appropriate, update the existing record if changes were made) AND then immediately proceed to replacing the variables with the assigned values throughout the document. We recommend that you always click Next unless you do not want the variables replace.

6.4 Replacing Variables with Values

When you have finished providing personal data in the right column of the Instant Database screen, press the **Next>>** button. This is where the "database" part of "Instant Database" comes in:

- Pathagoras will ask if you want to save the personal data as a new data record.



- If you plan to compose more documents for this customer or client (e.g., future letters, forms, contracts, pleadings, addendums etc.), you should say 'Yes.' When prompted, provide a meaningful and unique name for the record. (Typically, the client or customer's name, e.g., "Lastname, Firstname" meets this criteria.)

If you do click 'Yes,' then when you create another document for the customer or client containing some or all of the same variables, you need only drop down the Existing Data list (upper right side of screen) and locate the record. That is why we call it 'Instant Database.'

- Of course, you can also say 'No.' If you believe that the data is not necessary to keep, that should be your answer.

After your 'Yes' or 'No' response, Pathagoras will make the replacements. It should take only a couple of seconds.

6.5 Data Records

A 'Data Record' is a single or unique activity for a particular client, patient or customer. From a practical standpoint, in most cases it *is* a single client, patient or customer, but this should be further clarified.

A 'Data Record' is simply the pairings of variables and the values recorded onto the Instant Database screen and which is saved to a file on your computer. This data record can be reused with other documents containing the same variables, avoiding reentry of the same data.

Creating a new Data Record:

After document assembly ("ad hoc"):

Fully discussed under the [Scan](#)¹⁸⁸ sub-heading. Creating the Data Record after the assembly of the document is the 'classic' technique, but by no means a required technique. You can create a Data Record at any time.

Prior to document assembly:

Pathagoras contemplates that many offices may wish to input a client or customer's data independently of a document assembly session. So, a data input clerk who has no document assembly responsibilities can still create a Data Record for use by another staff member during a later document assembly session.

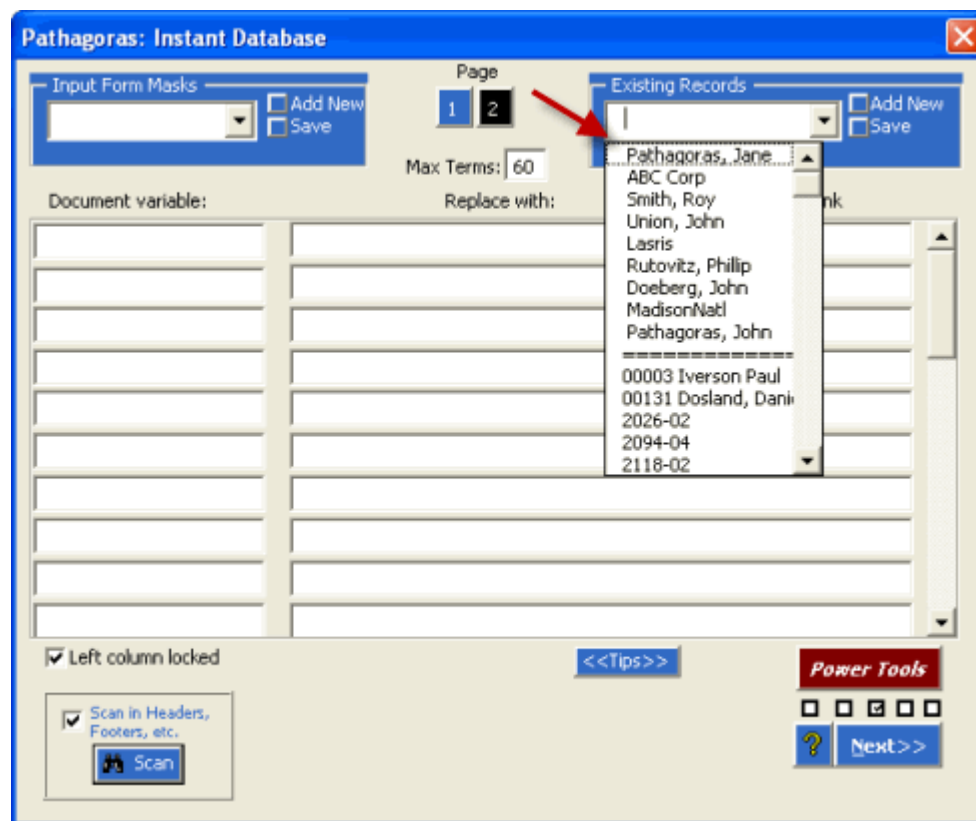
In order for this feature to work effectively, you must have previously created an 'Input

Form Mask' into which the personal data can be input. An Input Form Mask is simply an Instant Database screen which contains a complete list of the variables for which you wish to capture personal data. The mask becomes the vehicle by which personal data can be requested and then stored.

See [Masks](#)²⁶⁰ for the steps needed to create an Input Form Mask.

Using Data Records -- the 'database' part of "Instant Database":

Assemble a *second* document for a client or customer. Press <Alt-D> to display the Instant Database screen. Instead of scanning for [bracketed] variables or recalling a mask, click the Existing Records drop down list in the upper right corner of the Instant Database screen. Find and click the client or customer name you saved earlier. Make any changes to the data that may be appropriate. Click **Next>>**. That is all!



To call in an existing record, click on the Existing Records drop down list.

The original default location where Pathagoras stores Input Masks and Data Records is:

"C:\Program Files\Pathagoras\IDBS".

- You can easily navigate to this location using ordinary Word and Windows navigation techniques.
- You can easily change this location. Those who have purchased a network package may wish to point the default location to one on the server. See [IDB File Locations](#)²⁵⁰

There is no limit to the number of times that you can reuse a Data Record. There is no limit to the number of data records you can create.

The last used records (up to 9) appear in the top portion of the 'Existing Records' drop down list.

6.6 Augmenting an Existing Record

Let's say that you have an existing Instant Database record which has [Client Name] and [Client Address] saved and paired with "John Doe" and "123 Main Street".

You call up another document and that document contains variables called [Spouse Name] and [Spouse Address]. How do you add those variables (and the values you want to assign) to John Doe's record.

In many other programs, to add a variable that is not present in the 'base' variable list, you must first modify that base list to include the new variable. Not so with Pathagoras. (Now, to be fair, [Spouse Name] and [Spouse Address] are the kinds of variables that are part of the 'base' variable list. Editing the list to include these variables will not be likely. But what about [3rd Alternate Personal Representative]?)


Pathagoras' Instant Database system is unique in its ability to add variables to the database without requiring any modification of the underlying database structure. Whatever Pathagoras finds, it adds.

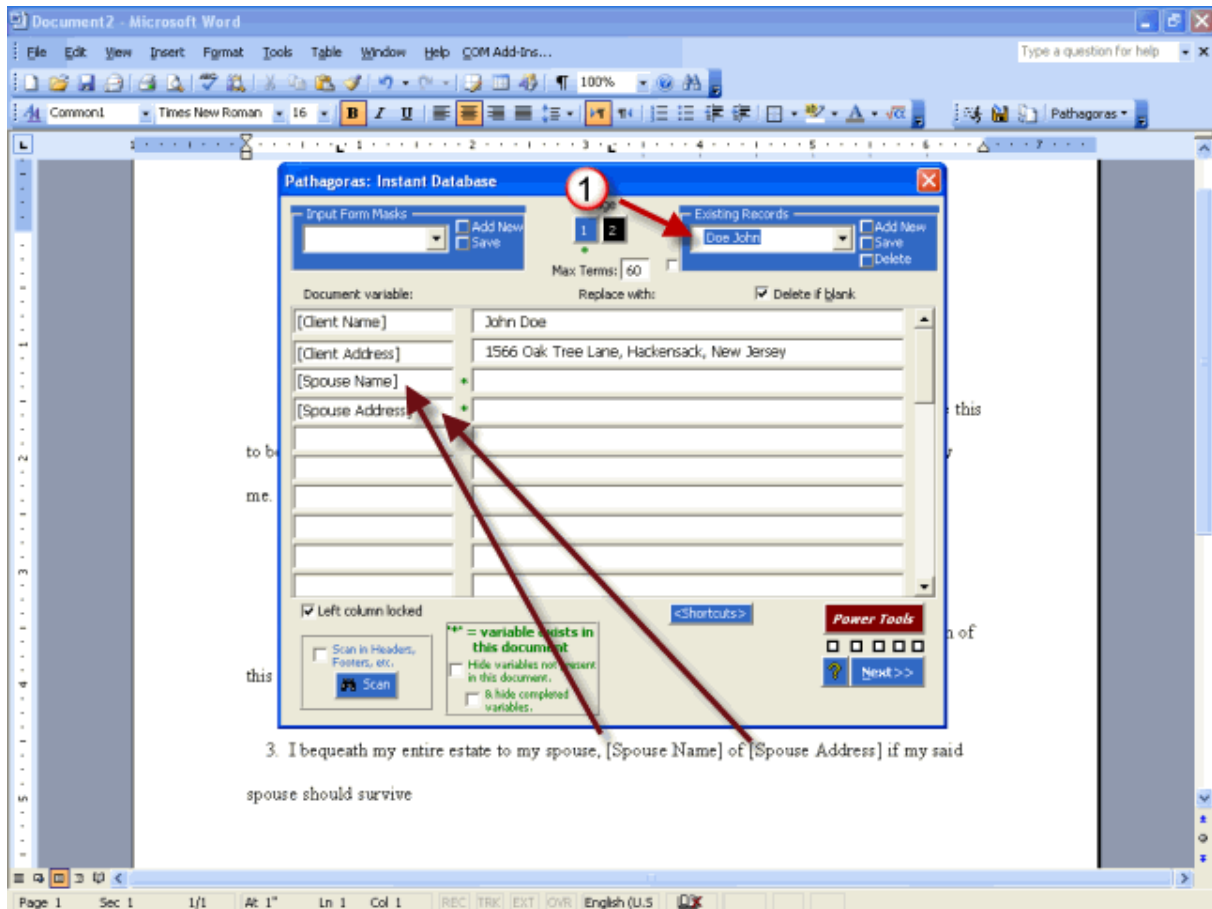
So, if you have variables from a mask or existing data record and want to combine them with additional variables from the on-screen document, it is a simple operation.

1. Just call up the Instant Database.
2. Display the client or customer's record.

Pathagoras will display the currently saved values and then will automatically scan the underlying document for the new variables.

3. The new variables are added to the 'bottom' of the list.

 The order of 'events' is critically important. It's 'recall record' first. If you scan first and then recall the record, you will simply overwrite the data from the initial scan. Further, if you completed the right column with values for the scanned variables, you will lose that work when you recall an existing record.



First select the record you want to use.
Pathagoras will then automatically locate and display
the 'new variables' at the end of the list.

6.7 Variables, Number of

Pathagoras starts with the default maximum number of variables set at 60. You may have, however, up to 999 variables.

- On a blank IDB screen, you will probably see '60' in the text box in the upper middle section of the screen, just beneath the 'page turner' buttons. This number will be higher if a mask or previously saved Data Record required more variables.
- You may increase this number in increments of 10 (all the way to 990) by manually typing a higher number (in multiples of 30) in the text box.

6.8 Variables, Length of

A variable can be any length.

That said, the best variables are short. Short variables are easier for the user to see and process when encountered in a document.

When a particular variable exceeds a certain length, it becomes unwieldy. There is no definitive size when this occurs, but Pathagoras suggests 40 characters as the 'max.' (Indeed, '40' is set as the default maximum variable length when you install Pathagoras)

Can a variable be 'too short'? Yes. When the replacement value is not clearly suggested by the variable, it is too short. The variable '[NOC]' (standing for, perhaps, Name of Client) is probably too short. Why? Few people (especially new staff just learning your systems) would know what that abbreviation means. ("[DOB]" may work because it somewhat has a universal acceptance as the abbreviation for Date of Birth.)

Similarly, "[Name]" is probably too short for most situations. Who's name is being called for? On the other hand, "[Name of Client]" is perfect. So would be "[Name of Our Client]", "[Client Name]", "[ClientName]", and a myriad of other choice. "[Name of our client as indicated on the first line of the Client Intake Sheet]" is 'legal', but too long for 'best practice.' (What about [Client's Name]? See the [discussion here](#)¹⁴⁰ as to why quotation marks and apostrophes are a 'bad idea' within variable names.)

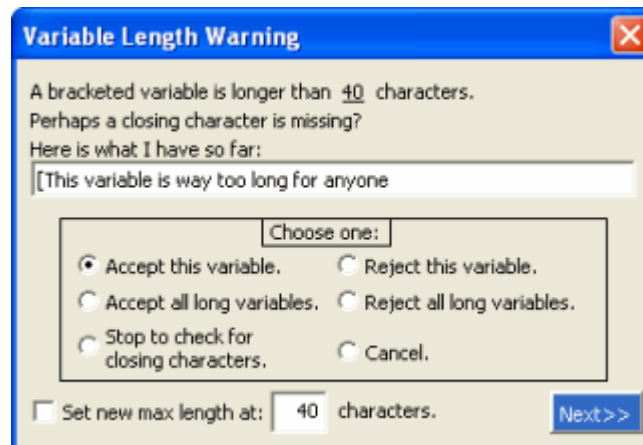
Q. I know that my clauses are all short, yet Pathagoras tells me otherwise. What's happening?

Two probabilities:

- Most likely you are missing bracket to close the variable. Since Pathagoras cannot tell where the variable ends, it assumes that the 'rest of the document' is the variable. Pathagoras reports this back to you via the screen shown below.
- Also likely: you have some explanatory text (perhaps from a legacy source) that was enclosed within the same bracketing set you use for your variables. Pathagoras cannot distinguish between such explanatory text and a variable that you have created.

Q. So what options are available to me?



- If you encounter a 'variable is too long' error during a document assembly session, Pathagoras will pop up this screen as it is scanning the document for variables:



Study the available options. Make a selection. The selections will hold for the remainder of the Scanning session. (Exception: if you set a new max length, that will permanently reset the value.)

Click Next>>.

- Regardless of the choices you make while in 'live' operation, you should rethink the underlying form and make changes as appropriate. If the source text is a missing bracket, you need to go into the source document to add it.

- If the problem is 'explanatory text' from legacy sources, decide if you need to retain that text. If you do, then open the source text and change the enclosing characters to something other than the one used to identify variables.
- If the variable is 'correct' but just longer than the default maximum, either reduce the size of the variable in the source text to the 'max length' or increase the default maximum length for variables. You would accomplish the latter in the [Instant Database Tools](#)  screen.
- You can tell Pathagoras to completely ignore variables that exceed the maximum length. This is also done in the [Instant Database Tools](#)  screen.

6.9 Variables, Special

When we think of variables, we typically think of a name, address, date, quantity, color, etc. But we seldom think of variables as entire documents.

But with Pathagoras they easily can be.

Examples:

A **real estate agent** or **property manager** might link to a folder containing the lengthy descriptions of properties under contract. Maybe the variable would be [Property Description].

A **technical writer** might link to long attachments that need to be appended or inserted at strategic locations.

An **architect** might link to drawings and tables and charts. (The target information need not be text. It can be can be charts or images contained within the target document).

A **bid managers** or **proposal drafters**, when preparing Request for Proposal or a Bid package, you may wish to insert a special condition of the anticipated contract (perhaps it is not part of your standard package, but rather was drafted especially for the vendor or supplier or situation, but you want to save it out as a document). Maybe the variable would be [Special Conditions]

A **doctor** might link to an appropriate medical pamphlet or brochure that can be inserted as an appendix to a letter to a patient.

A **psychologist** might link to a folder containing detail information about a particular test or protocol.

A **lawyer** designing an **estate plan** might wish to insert a page or two discussing transfers of specific assets mentions in a Living Trust. The variable might be called [Transfer Instructions]

When the assembled document is personalized using Instant Database, the document that has been paired with the variable will be automatically inserted in place of the variable.

This tool can also be used to assemble a document (or portions thereof) when the various portions come from different folders. (Document assembly is dependent upon all clauses being within a single folder or glossary. Here, you can assemble a document pulling distinct chapters out of separate folders.)

BONUS: If a variable such as [Special Conditions] or [Property Description] or [Transfer Instructions], etc., can be one of several documents, Pathagoras lets you to link

the variable *to an entire folder of documents*. The documents in the designated folder as displayed as a drop down list next to the variable (just like a multiple choice variable would display) and you would choose the desired document from among those displayed.

The next two sections discuss your options. The first discusses how to assign a <<filename>> to a variable. The second discusses how to assign a folder.

To replace a variable with a document, you need only type the name of the document to the right of the variable. Enclose the name within << and >> markers so that when it is inserted into the document, it will be automatically recognized and processed by Pathagoras. (See _____ for more on the <<FileName>> function. So, if you know the name of the document you want to substitute for the variable, you can enter it into the right column in two ways:

1. Type the fully qualified name of the document. (If you leave off the doc extension, don't worry. Pathagoras will add it.)
2. If the document is stored in your Super Folder, just type the name of the document. (If you leave off the doc extension, don't worry. Pathagoras will add it.)
3. Navigate. This is really cool. Press down the shift key and then click the mouse in the IDB field. A menu will appear. Follow the prompts and navigate to, and select the document you want the variable to represent.

6.10 Validating Variables

"Validating a Variable" is geek for making sure that the data input is consistent with the request made and in the right format.

Pathagoras cannot yet make sure that you put in only numbers when, let's say, a phone number is requested, but you can insure that the formatting of Dates, Telephone Numbers and Social Security numbers is proper before the replacements are made from whatever is input by the user into a particular style that you want. (e.g., "1/28/13" becomes "January 28, 2013" with a simple double-click on the field.)

Pathagoras let's you set the default formats for three common items: Dates, Phone Numbers and Social Security numbers. No matter the initial value input by the user, a double click on the text box in which the value is input transform it into the proper format.

The requirements are simple:

- (a) A key word must exist in the variable name so Pathagoras knows what kind of data you are working with.
- (b) The right (value) side must contain at least an 'acceptable value'. (If you want to convert a telephone number that contains letter, or an insufficient number of digits at the right, Pathagoras won't know what to do with it.) More on this immediately below:

Dates: (a) The key word "Date" or "DOB" must exist in any fashion within the variable name. (b) An acceptable value is anything that appears as a date. The 'return' will be the default date format you selected for other Word functions.

Telephone Numbers: (a) The key word "phone" or "fax" must exist in any fashion within the variable name. (b) An acceptable value is any 10 numbers, with or without parentheses and hyphen. The 'return' will be the default telephone format you selected in the Instant Database


Settings and Options screenage.

Social Security Numbers: The key word 'SSN' or 'Social Security' must exist in any fashion within the variable name and the value must be all numbers and 9 digits long. The 'return' will be in the format XXX-XX-XXXX.

6.11 Math and Date Math

Simple math can be performed using any existing Instant Database variable which has been assigned a numeric value. Create a formulas and assign its result to another variables.

Pathagoras can perform addition, subtraction, multiplication or division on up to four operands.

 This feature presupposes that you will be preparing an [Instant Database mask](#) to house the formulas. If you want to place the formula in-line with the text of the source document, see this page.

Example: Let's say that you have a document containing, among others, four variables that you want to use to calculate a total price. Perhaps they are called [Unit Price], [Quantity], [Discount] and [Total Price]. These variables are contained in a mask designed for this and similar 'price quoting' documents.

The formula in the mask adjacent to [Total Price] might look like one of these:

[Unit Price] * [Quantity] - [Discount] *(if the variable discount is a flat amount)* OR

[Unit Price] * [Quantity] * [Discount] *(if the variable 'discount' is a percentage)*

OR

[Unit Price] * [150] * .25 *(you can use 'real' numbers if you wish, with or without brackets).*

[Unit Price] * 150 * .25

In actual operation, you would assemble the document for the customer and display the appropriate IDB mask. You would complete the 'personal variables' (name, address, etc.) and insert the appropriate values for each of the first three variables (unit price, quantity and discount) which feed the formula.

When Pathagoras detects a formula (the presence of math symbols is the give away) in the personal data side of the IDB screen, it will display a small red button between the two columns. (Similar in size to the 'date picker' button.) Click the red button and the formula is processed.

If any figure that is part of the formula changes, the [Total Price] can be recalculated by two successive clicks of the calculator button. The first click restores the formula. The second click recalculates using the new values.

You can have a total of 4 "operands" (and therefore up to 3 operators).

Here is another example:

Pathagoras: Instant Database

Input Form Masks: **Order Form** [Add New] [Save] [Delete]

Page: 1 2 Max Terms: 60

Matter Data: [Add New] [Save]

Document variable: Replace With: ☒ Delete if blank

[Customer Name]	Full Customer Name (e.g., "Great Foods, Inc.")
[Customer Street Address]	Address line 1 (e.g., "123 Main Street")
[Customer Address 2]	Secondary address (e.g., "Suite 45")
[Customer City, State Zip]	Customer City, State and Zip Code
[Product Ordered]	Describe product ordered
[Quantity Ordered]	Type quantity of product ordered (numbers only, e.g., "100")
[Price per Item]	Type unit price of product ordered (numbers only, e.g., "15")
[Total of Order]	[Quantity Ordered] * [Price per Item]
[Date of Order]	Type date of order, or click green button to select from calendar.

A mask called **Order Form** was created for quoting prices. It contains a simple formula to calculate the value for the variable called **[Total of Order]**.

Pathagoras: Instant Database

Input Form Masks: **Order Form** [Add New] [Save] [Delete]

Page: 1 2 Max Terms: 60

Matter Data: [Add New] [Save]

Document variable: Replace With: ☒ Delete if blank

[Customer Name]	Great Foods, Inc.
[Customer Street Address]	123 Main Street
[Customer Address 2]	Suite 45
[Customer City, State Zip]	Jacksonville, FL
[Product Ordered]	Widgets
[Quantity Ordered]	25
[Price per Item]	47
[Total of Order]	1,175
[Date of Order]	Date of order, or click green button to select from calendar.

When the mask is completed, and after values have been assigned to the base variables upon which the formula depends, click the red **Calculate** button. If a base figure changes, click the button twice to recalculate the value.

The allowable 'math operators' are:	
Addition: "+"	Subtraction: "-"
Multiplication: "*" or "x"	Division: "/"

DATE MATH:

Date math (addition) can be performed from the Instant Database screen following the same setup styles as used with the above 'plain' math examples. The formula is written like this:

$$=[\text{Base Date}]+[4 \text{ months}]$$

where [Base Date] is any date variable in the mask (the variable must contain the word 'date', but it can be in any form). The time interval additur can be any number, followed by a space, and concluded with either 'days', 'weeks', 'months' or 'years' (or the singular).

Example:

The screenshot shows the 'Pathagoras: Instant Database' application window. It features a 'Formula' tab and a 'Matter Data' section. The main area displays a list of document variables and their corresponding values. The variables are: [Customer Name], [Customer Street Address], [Customer Address 2], [Customer City, State Zip], [Product Ordered], [Quantity Ordered], [Price per Item], [Total of Order], [Date of Order], and [Delivery Date]. The values are: Great Foods, Inc., 123 Main Street, Suite 45, Jacksonville, FL, Widgets, 25, 47, 1,175, March 22, 2010, and [Date of Order] + [3 weeks]. A red arrow labeled '1' points to the green 'Calendar' button next to the [Date of Order] variable. Another red arrow labeled '2' points to the red 'Calculate' button next to the [Delivery Date] variable.

Document variable:	Replace With:
[Customer Name]	Great Foods, Inc.
[Customer Street Address]	123 Main Street
[Customer Address 2]	Suite 45
[Customer City, State Zip]	Jacksonville, FL
[Product Ordered]	Widgets
[Quantity Ordered]	25
[Price per Item]	47
[Total of Order]	1,175
[Date of Order]	March 22, 2010
[Delivery Date]	[Date of Order] + [3 weeks]

(1). Set the 'base' date by clicking the green 'Calendar' button. (You can also manually type in the date).

(2). Click the red 'Calculate' button on the target line. You should see the result as shown below:

Document variable:	Replace With:
[Customer Name]	Great Foods, Inc.
[Customer Street Address]	123 Main Street
[Customer Address 2]	Suite 45
[Customer City, State Zip]	Jacksonville, FL
[Product Ordered]	Widgets
[Quantity Ordered]	25
[Price per Item]	47
[Total of Order]	1,175
[Date of Order]	March 22, 2010
[Delivery Date]	April 12, 2010

If you change the 'base date,' click the calculate button twice.
 The first click will restore the formula (in case you want to modify it).
 The second click will perform the calculation on the new assumptions.

Date "difference" math (how many days, weeks months or years have passed between two known dates) can be performed from the Instant Database screen, again using a mask to save the formulas for repeated use.

The same setup styles as shown above is used. Sample variables, a sample formula and screen shots, are provided below:

[Base date]

[End date]

[Span, in Months] =[Begin Date]-[End date] (the formula beginning with '=' is placed in the right column)

[Age in Years] =[Begin Date]-[End date] (the formula beginning with '=' is placed in the right column)

The screenshot shows the 'Pathagoras: Instant Database' window. At the top, there's a blue header bar with the title and a close button. Below the header, the interface is divided into several sections. On the left, there's a section for 'Input Form Masks' with a dropdown menu showing 'aaaTestDate' and buttons for 'Add New', 'Save', and 'Delete'. To the right of this is a 'Page' section with buttons for '1' and '2', and a 'Max Terms' field set to '60'. Further right is an 'Existing Records' section with a dropdown menu and buttons for 'Add/Move' and 'Save'. Below these sections is a 'Document variable:' section with a red square icon and a 'Guide to complete the mask' section with a checked 'Delete if blank' option. The main area is a table with two columns: the left column contains variable names in brackets, and the right column contains their values. The variables are '[Base date]', '[End date]', '[Span Months]', and '[Age in Years]'. Their values are 'January 1, 1950', 'March 1, 2013', '= [Base Date]-[End date]', and '= [Base Date]-[End date]' respectively. At the bottom left, there's a 'Left column locked' checkbox and a 'Scan' button. At the bottom right, there's a '<Shortcuts>' button, a 'Power Tools' button, and a 'Next>>' button.

Document variable:	Guide to complete the mask
[Base date]	January 1, 1950
[End date]	March 1, 2013
[Span Months]	= [Base Date]-[End date]
[Age in Years]	= [Base Date]-[End date]

This screenshot shows the same 'Pathagoras: Instant Database' window, but with the calculated results for the form masks. The values in the right column are now 'January 1, 1950', 'March 1, 2013', '758', and '63'. The 'Span Months' value '758' is circled in green, and the 'Age in Years' value '63' is circled in blue. The variable names in the left column are also highlighted: 'Months' in 'Span Months' is green, and 'Years' in 'Age in Years' is blue.

Document variable:	Guide to complete the mask
[Base date]	January 1, 1950
[End date]	March 1, 2013
[Span Months]	758
[Age in Years]	63

There are 758 months between the two dates, which converts to 63 (whole) years. The numeric result of number of days, weeks, months or years is controlled by the existence of the appropriate term "day(s)", "week(s)", "month(s)" or "year(s)" contained in the variable at the left.

6.12 Multiple Choice Selector

"[Chocolate/vanilla/strawberry]" is a multiple choice item. So is [*states*]. When a multiple choice list is encountered by Pathagoras, it parses the various options and presents the various

choices in a dropdown list. Typically, you would drop down the list, select a single value from the list, and move on. But what if the answer requires multiple selections from the list. Can that be done? Of course!

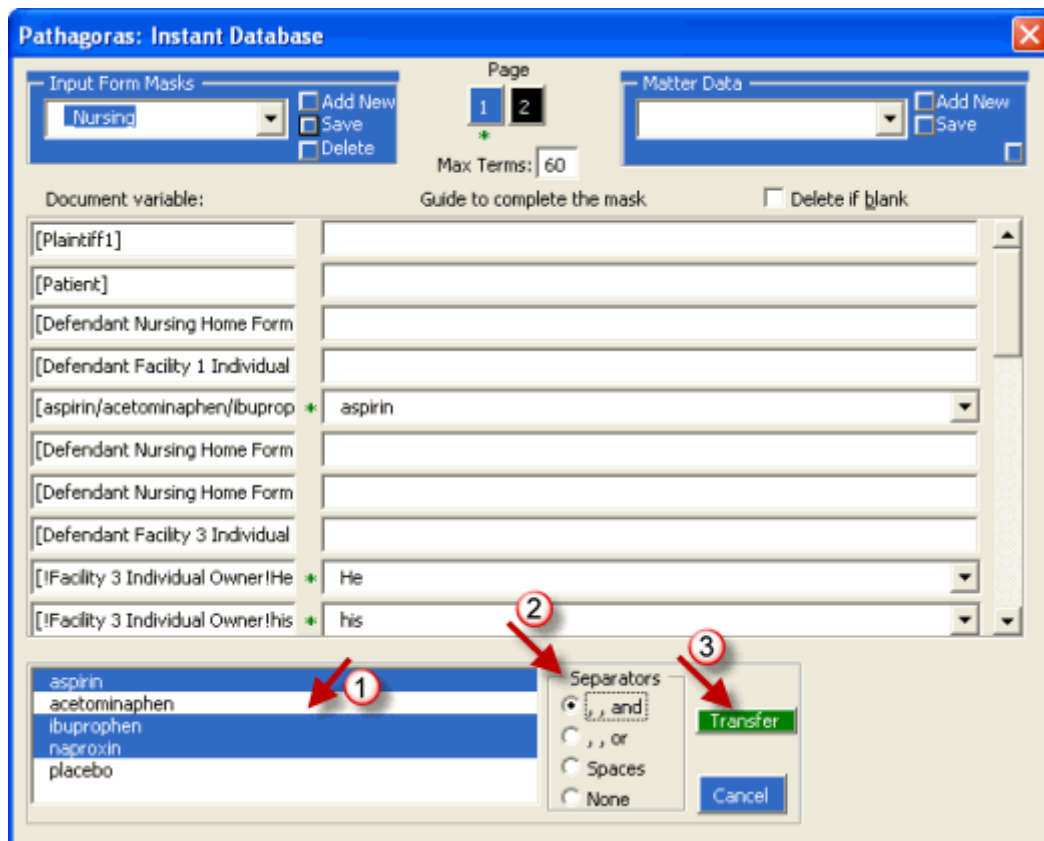
When you click inside of a drop down list containing multiple choices, a MultiSelect button will appear inside the row at the right. (NOTE: The first time you try this, you may have to tell Pathogoras to display the MultiSelect button. Look for the checkbox that appears just beneath the last 'replacement entry' location titled "MultiSelect Button Visible. (It will appear only when you are within a multiple choice dropdown.) Click it.)

If you desire to choose more than one item from a list, enter the text box containing the list and, if it is not already checked, check the 'MultiSelect button Visible' box.

A new button will appear within the boundary of the current list (and any other list you may enter) called MultiChoice. Click it.

When MultiSelect box is checked, a MultiSelect button appears.

A new section of the screen opens up containing a listing of all of the choices. Make appropriate choices from the list. (Regular Windows list-choosing techniques -- shift-click and ctrl-click -- work here.) Next, if desired, choose a method by which the terms will be 'separated' and 'connected' (by commas, followed by an 'and' or 'or', etc. or 'No connectors.') Press the Transfer button and the properly formatted text will be transferred into the Replacement text area.



Select choices (Ctrl- or Shift- click techniques work).

Click a separator, and then click Transfer. That's it!

In the example above, when <Transfer> is clicked, the replacement text will become "aspirin, ibuprophen and naproxin".

You can change the 'default' separator via the 'All Settings' | 'Miscellaneous Settings' screen. See [Separators and Connectors](#) ⁵⁷⁸ for more information.

6.13 Embed Variable Names

Typically, when you run the Instant Database and replace variables with personal text (i.e, [Client Name] with "John Doe", the variable is simply replaced and all remnants of the previous text are gone. (This is so with all Word search and replace exercises.) This is the fastest and most efficient way to get from draft to personalized document.

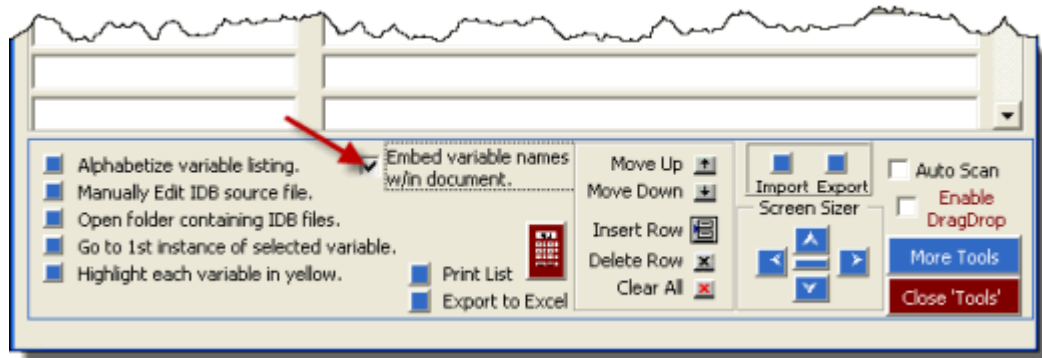
However, there are times where you want to return to the document to make corrections. The classic method is to perform a standard Word search and replace routine on the document. But there may be situations where the term you are seeking to correct occurs naturally in the document in places other than those which you wish to change. In such cases, you have to enable the one at a time replacement method, and that can be a bit 'clunky.'

Beginning with Pathagoras version 2011, you can tell Pathagoras to embed the original variable names as a physical part of the the document. When this feature is enabled, you can correct misspellings or make other changes to the replaced text in (and only in) those locations you desire. To do so, simply recall the document and activate Instant Database. The embedded IDB record will be displayed. Note the changes you wish in one or more variables. When you press the Next button, all old values deriving from the original [variable] are updated in the

document. The IDB record itself is updated and re-embedded.

There is a bit of processing time added to the replacement process as the embedding takes place, but for short documents, it is hardly noticeable.

To embed the variable names, activate the Instant Database screen and click the red **Power Tools** button (lower right corner of screen). When the new section appears, check the 'Embed variable names w/in document' box.



The Instant Database 'Power Tools' section. "Embed Variable Names" checked.

A possible 'abuse' of this new feature (and only you can decide how you want to implement it) will let you recall an existing document which contains 'replaced text' (i.e., the IDB has been run against the document), and change the values of all of the variables. (Here, we are not talking about typos or an address change, but an entire new customer/client/patient). You can also apply a completely new IDB record to the document. All fields will be appropriately replaced with an entirely new data set.

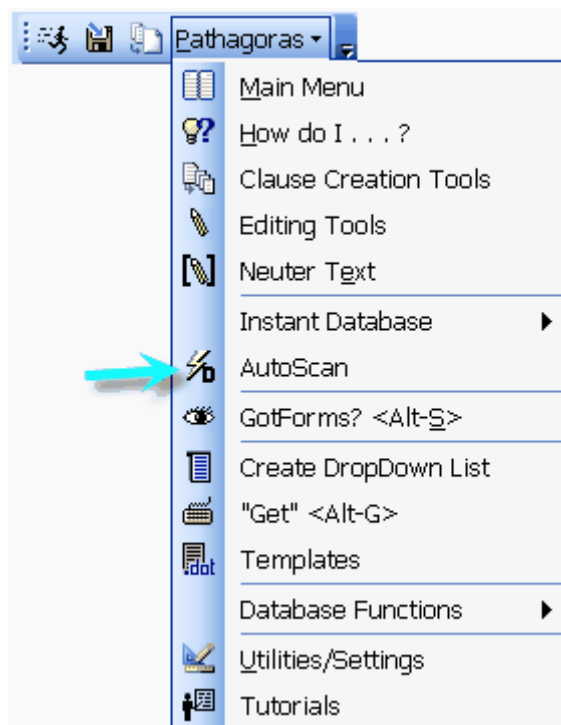
I say 'possible abuse' because if there is one thing that I have worked hard to protect against is a user reusing existing personalized text as a base for creating new documents. (You know -- the way you used to create documents before Pathagoras.) When the form is used in this way, there is a real risk that the user will forget to change all the variables that need changing. An old value might be accidentally left in. So, just like with pre-Pathagoras life, the resulting Will might give property to someone other than the client's spouse because the user forgot the replace the 'original' [SpouseName] value with the 'real' value.) Not only is it embarrassing, but it is a potential breach of confidence.

Alas, I cannot write the routine to protect against everything, so if you choose to use it in this fashion, just beware.) It is our advise that you tell you end users to use this feature only to make slight changes from the original.

6.14 AutoScan

There are two ways that you can get Pathagoras to automatically scan your documents for any bracketed variables.

- **AutoScan button:** You can click the AutoScan button found in the Pathagoras drop down features list. This performs two steps. It activates the Instant Database screen and scans the document with additional 'clicks.'



- **AutoScan setting:** You can set both the 'display' and 'autoscan' steps to the <Alt-D> hot-key. To set this feature,
 1. Display IDB.
 2. Click the Power Tools button.
 3. Check the AutoScan checkbox.

6.15 Other IDB Functions, Features and Tools

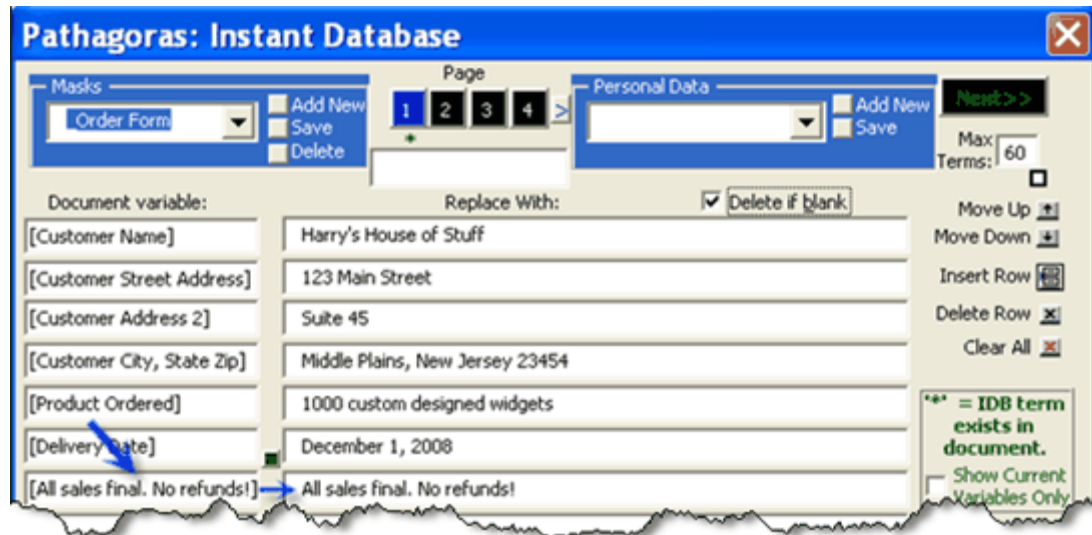
There are a few more features of the IDB system of which you should be aware that have not been discussed elsewhere:

- **Double Click:** Double clicking on a variable name in the IDB mask will cause the variable name (but not the brackets) to be transferred to the right side of the form. This is a 'poor man's way' to optional text. To 'keep' the text, double click it so that it stays in the document. (The actual operation is that it replaces itself with itself.) To delete the text, just leave the 'replacement text' side blank.

Example: Let's say that you are preparing a cover letter to accompany a purchase order. In some cases, the purchase is final, while in others a product return is allowed. To meet this situation, you might place a variable in the source document that reads "[All purchases final.]"

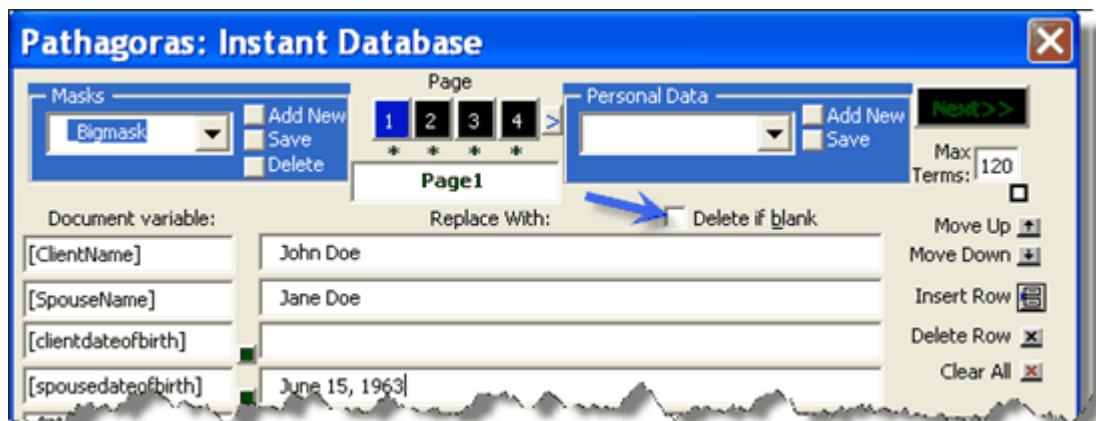
When it is time to personalize the document, that variable (along with all other document variables) will be presented on the Instant Database screen. If this is one of those sales

where no return is allowed, the user could double click the "[All purchases final.]" variable and it will be transferred to the right as the replacement text. When you click the Next button (telling Pathagoras to make the replacements), "All purchases final." will appear in the final letter. (Of course, if refunds are allowed, the operator would simply leave the replacement text blank. When Next is pressed, the variable will be deleted and the proper message will be communicated. (In actuality, the variable is replaced with 'nothing,' but the result is the same.)



Double click on variable transfers text (without brackets) into replacement column.


- **Delete/Retain Variables:** You may wish to complete a document on which you are working with the replacement text you current have and complete the remaining variables at a later time. Pathagoras lets you do this. Simply uncheck the "Delete if blank" box above the first line of the replacement text column. By the same token, if you wish to delete a variable from the editing screen if its value is 'blank', check the 'Delete if blank' box. (If the 'blank' variable appears on a line by itself, the paragraph marker closing the line is also removed so no empty space remains.)



Uncheck 'Delete if blank' box to retain any variables for which data is not currently available.

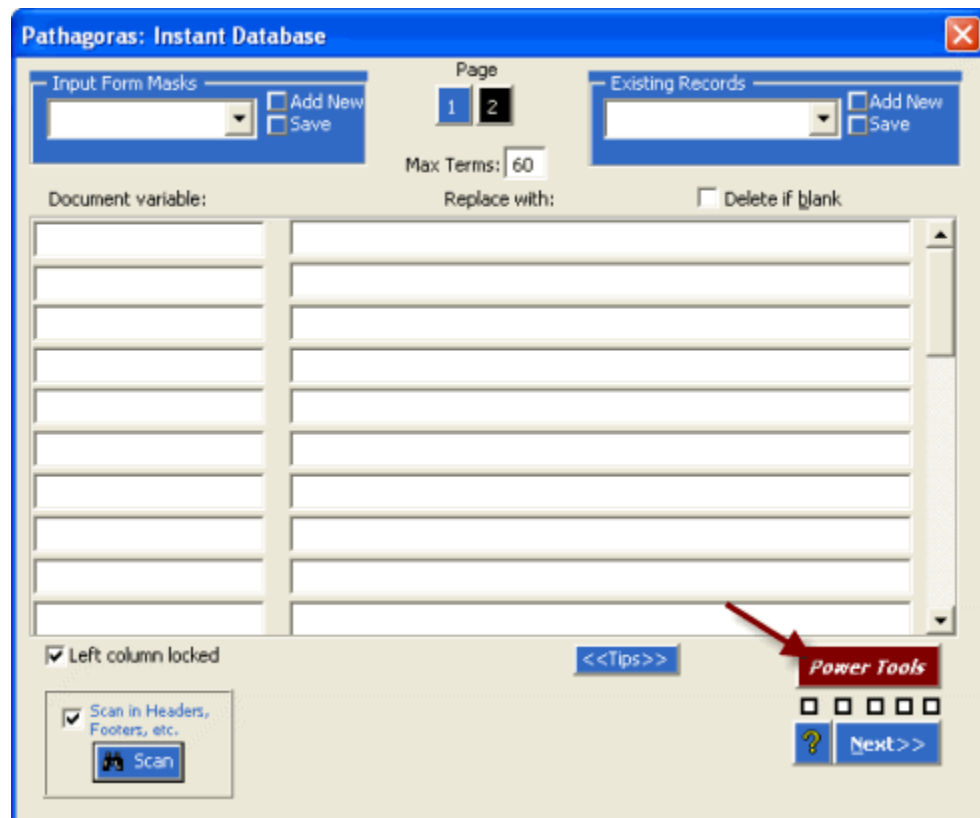
- **Copy variable, brackets and all:** Two bullets up, we mentioned that double clicking on a variable will transfer the content of the variable to the 'value' (right) side of the screen,

without the brackets. Let's say that you (1) want to preserve a particular variable, but want to delete all other 'blank' variables (the bullet immediately above). To accomplish this seemingly contradictory feat, just copy the variable onto itself. The value is not 'blank' so it won't be deleted, but the variable also remains in the document for later completion. A 'shift-double click' on the variable accomplishes this. Unlike the action two bullets up, the brackets are preserved.

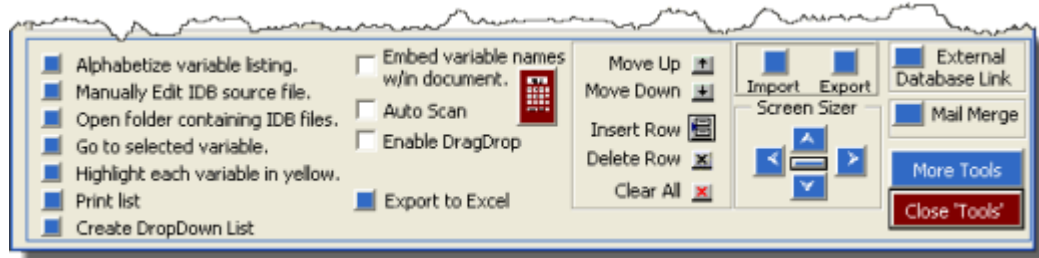
 There is another kind of 'Preserve Variables' function available in Pathagoras. [Click here to read about that alternative.](#) ²⁰⁶

6.16 Power Tools

A wide range of tools and additional features reside behind the <Power Tools> button on the IDB screen.



Pressing it reveals at screen bottom a collection of editing tools and other resources:



Here is a brief explanation of what each button controls:

- **Alphabetize** entries: Some users who wish to create an Input Form Mask want the variables displayed in alphabetical order. Use this button to achieve that result.



The feature can be used to quickly detect spelling errors in your source documents involving similarly named variables. Let's say you have typed what you intended to be the *same* variable, but used a slightly different 'spelling' of the variable at various locations in the document. (E.g., [ClientName], [Client Name], and [Client_Name].) You can quickly determine whether all variables are properly 'spelled' by scanning the document and 'Alphabetizing' the results. Using Power Tool button 'Go to 1st instance of Selected Variable.,' you can quickly locate the 'offending' variable(s) and make the appropriate corrections.




When does '100' come before '2'? When you are sorting numbers alphabetically (which is how it is done in the Instant Database):

Alpha-sorting of numbers is always done by the first character of the number, then the second, then the third, etc. It is identical to the way we alphabetize with letters.

So if you want 'numeric' variables to sort 'alpha-numerically', you need to pad the 'shorter' numbers with an appropriate number of '0's to make the shorted number the same length at the longest. In the above example, '100' may come before '2', but it will definitely come after '002'.

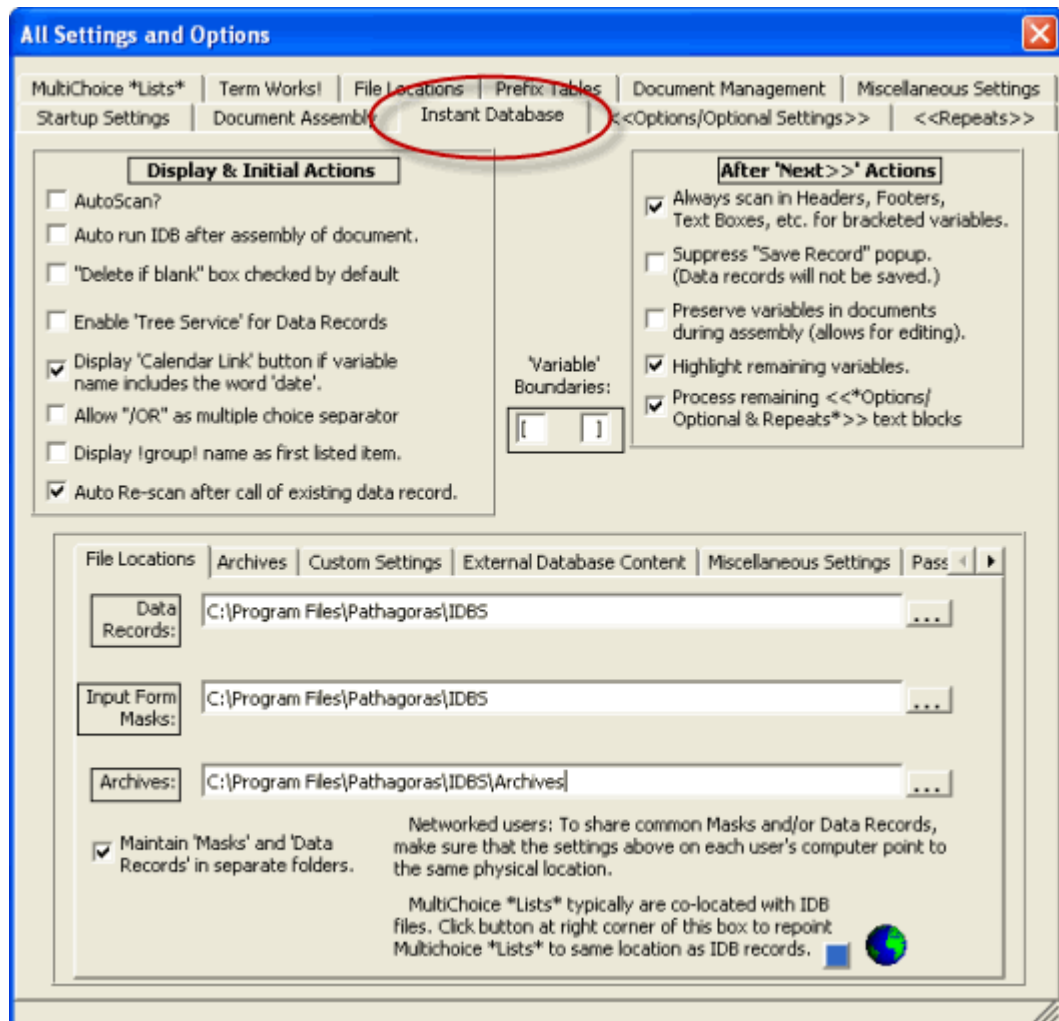
- **Edit Source File:** Should you even need to access the actual source record (not recommended), click the '**Manually Edit IDB source file.**'
- **Display Source Folder:** Likewise, should you ever want to look at (for editing, or just edification) all of the Instant Database records, click '**Open Folder.**'
- **Highlight Variables:** If you want the variables to 'jump out' on the screen, click '**Highlight**' to paint each variable in yellow. The highlighting is 'on-screen' only. It does not print.
- '**Go to 1st instance of selected variable**' is self-explanatory. Just make sure that the cursor is on the proper field.
- Clicking '**Create DropDown List**' will place all variables in the left side of the Instant Database screen into a drop down list in the toolbar area of the Word screen. With this list, you can point-and-click those variables into a document under construction. Speed up the creation of source documents, and avoids spelling errors during the placement of variables. You can refresh this list as you add new (different) variables to your document.

See [IDB Drop Down List](#)  for more on this topic.

- You can **Insert** a **List** of all variables that are displayed in the current screen. You are given three options. You can insert the list into (1) the current document or (2) into a new document. Either of these documents can easily be made into a client or customer intake sheet. Thirdly, you can insert the variables into your ['Variables' DropDown List](#) .
- You can **Export** the current display of variables (and values, if provided) to an Excel spreadsheet. You can edit the data there. The result will be immediately usable by Pathagoras for Word.
- **Calculator:** Pathagoras provides a simple **calculator** for your use. Press the red calculator button to activate this feature. The calculator will overlay the IDB screen. The value obtained using the calculator can be inserted into the last selected 'value' box with a single click.
- Click the **'AutoScan'** checkbox to tell Pathagoras to automatically scan the document for [bracketed variables] whenever the IDB screen is called up.
- **Edit Current Display:** The stand alone boxes toward the center of the Power Tools strip allow you to manually edit the Instant Database display. You would use this feature if you were creating (or editing) an Input Form Mask. You can move rows, insert blank rows and delete existing one.
- You can **Import** or **Export** IDB records to share with external users or external programs.
- **Resize Screen:** If the Instant Database screen is not wide enough to show the full name of the variables, or you wish to show more variables than the default '10' you can widen the screen or make it taller by clicking the appropriate Page Sizer buttons. To **lock** the resized screen, click the small bar which the Sizer buttons surround. (You can also change the width of the two dropdown boxes, and the default number of records each can show by making changes in the Miscellaneous Settings. This is found under the "More Tools and Settings.")
- **Change brackets:** If you prefer to use characters other than the default '[' and ']' (square brackets), change them here. You can use any combination of one or two characters except '<<' and '>>'.
- **External Database Link:** Pathagoras can draw data from other data sources. See External Database for more information and setup instructions.
- **Mail Merge:** Pathagoras can create several copies of the same form letter that you want to send to multiple addressees. See Mail Merge for more information.

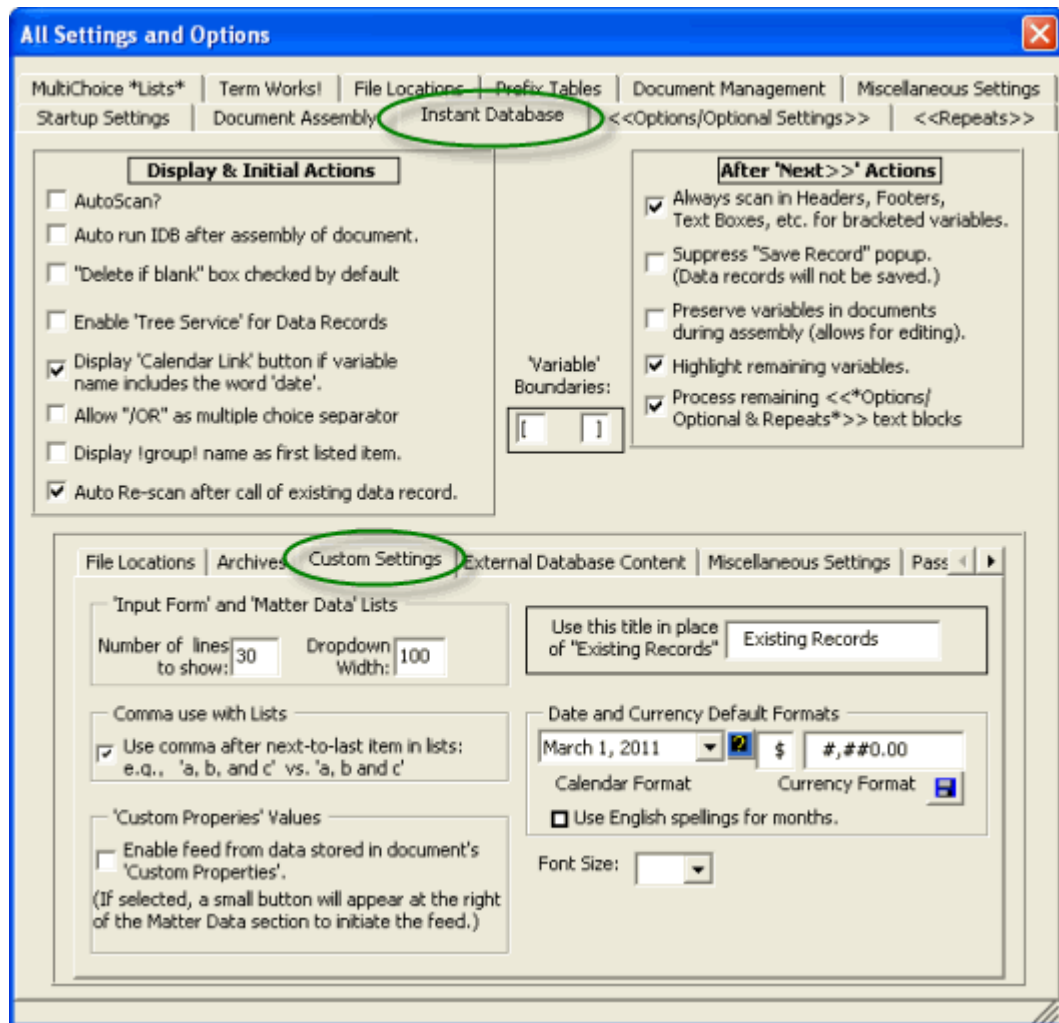
6.17 More Tools and Settings

There are a myriad of settings possible for your IDB module. They are available via the *Instant Database /Instant Database Settings and Options* selection found in the Pathagoras drop down features list. (This same screen can be accessed by clicking 'Power Tools' and then 'More Tools' from the Instant Database screen.)



Instant Database Settings and Tools.

Note the various categories of the display. Your Instant Database is practically infinitely customizable. The primary "Display & Initial Actions" settings are set out at the left of the screen. The "After 'Next>>' Actions" are provided at the right and a wide variety of other settings to customize Pathagoras for your particular office needs are made available in the bottom section of the screen.



Instant Database Tools (Custom Settings)

The various options and switches found throughout this screen allow you to highly personalize the IDB system to your needs. Here are several IDB settings which may be of particular interest:

In the Display and Initial Actions section:

- **AutoScan:** Pathagoras will scan the underlying document for variables without your having to press the Scan button. (If you frequently call in records from you list of existing records, you will probably *not* want this option checked. It is a bit of a time waster for Pathagoras to scan the document only to have you then select a record from the list, which essentially overwrites the scanned for records. But see the 'ReScan' feature below.)
- **AutoRun IDB:** After the assembly and processing of the document, Pathagoras is automatically activate the Instant Database screen and scan the resulting document for variables without any action on your part.
- **AutoReScan:** After you call a record from the Existing Records list (upper right corner of IDB screen), Pathagoras to rescan the document for variables in the underlying document that are not in the record, and if any are found, will add them to the bottom.

This allows you to continuously update a client/customer's record, adding new variables as they are 'discovered' in new document.



Note: An AutoReScan on long documents with many variables, can take several seconds. If you know that the ReScan will not discover more variables, you can temporarily disable the AutoReScan setting when recalling a record from the list of records in the Instant Database screen. Press Shift-Click on the record name (as opposed to a simple click).

- **Connectors:** You can change the 'structure' of a string of values from "apple, banana and strawberry" to "apple, banana, and strawberry." (Note the comma after the penultimate element.) This is a personal style choice.
- **'Legal' variable length:** Pathagoras 'gives up' on identifying text as a variable when it exceeds a certain length. 40 is the default 'max'. If you like to provide 'explanatory text' within the same type of brackets you use for your variables, you can set Pathagoras to exclude that explanatory text by changing the 'give up the search' length of variables.
- **Variable boundaries:** You can change the characters used to denote variables. Square brackets are the default, but you can use anything so long as it does not conflict with the 'simplified options/optional text characters, which are show as well on this screen.
- **Preserve Variables:** In the 'typical' Instant Database session, the [bracketed variable] is overwritten by the replacement text. 'Preserving the variable' keeps the original variable as hidden text. This in turn allows you recall a previously assembled document, call up the Instant Database, correct a spelling error, change an address, etc., and then 'update' the document. Read more at [this link](#)²⁰⁶.
- **Highlight Remaining Variables:** If, after running Instant Database against the document, you did not complete all variables, and you chose not to delete uncompleted variables, you can ask Pathagoras to highlight those that remain. This can serve as a visible reminder that there is still work to be done.
- **Tree Service:** Instead of storing all data records in a single folder, you can create sub-folders of your records, each sub-folder perhaps representing a major client or customer, or perhaps a major practice area. Individual data records can be stored into (and recalled from) those sub-folders.

Other Tools and Tabs: Click the various tabs along the top of the screen and other important information and data regarding the following topics will be displayed.

- **File Locations:** See where your IDB records are being stored. Or easily change the location where they are being stored. Some offices wish to share a common folder for all IDB records. In such case, point all users to the same location. Other offices wish to maintain separate files. In such case, make sure that all users point to separate location.
- **Variables:** Not really a 'tool,' but a brief lesson of how to create and use variables within the system.
- **Matching:** An advanced function. Match IDB variables with data from an external database (e.g., TimeMatters®, dBase®, Act®, etc.) and use the values from those databases as if you had manually typed them into your IDB.
- **Archives:** If your IDB Data Records are becoming too numerous, save them out to an

archive folder. Quickly restore one or more record if needed.

- **Custom Settings:** Under this tab, you can change the width and depth (number of records displayed at one time) in the two drop down lists on the page. You can direct Pathagoras to immediately print your document to the default or to a selected printer. (This latter feature can be viewed both as a convenience feature and as a security feature. By requiring that the assembled document be printed after assembly, there is some assurance that it has not been modified by the end-user or others.)

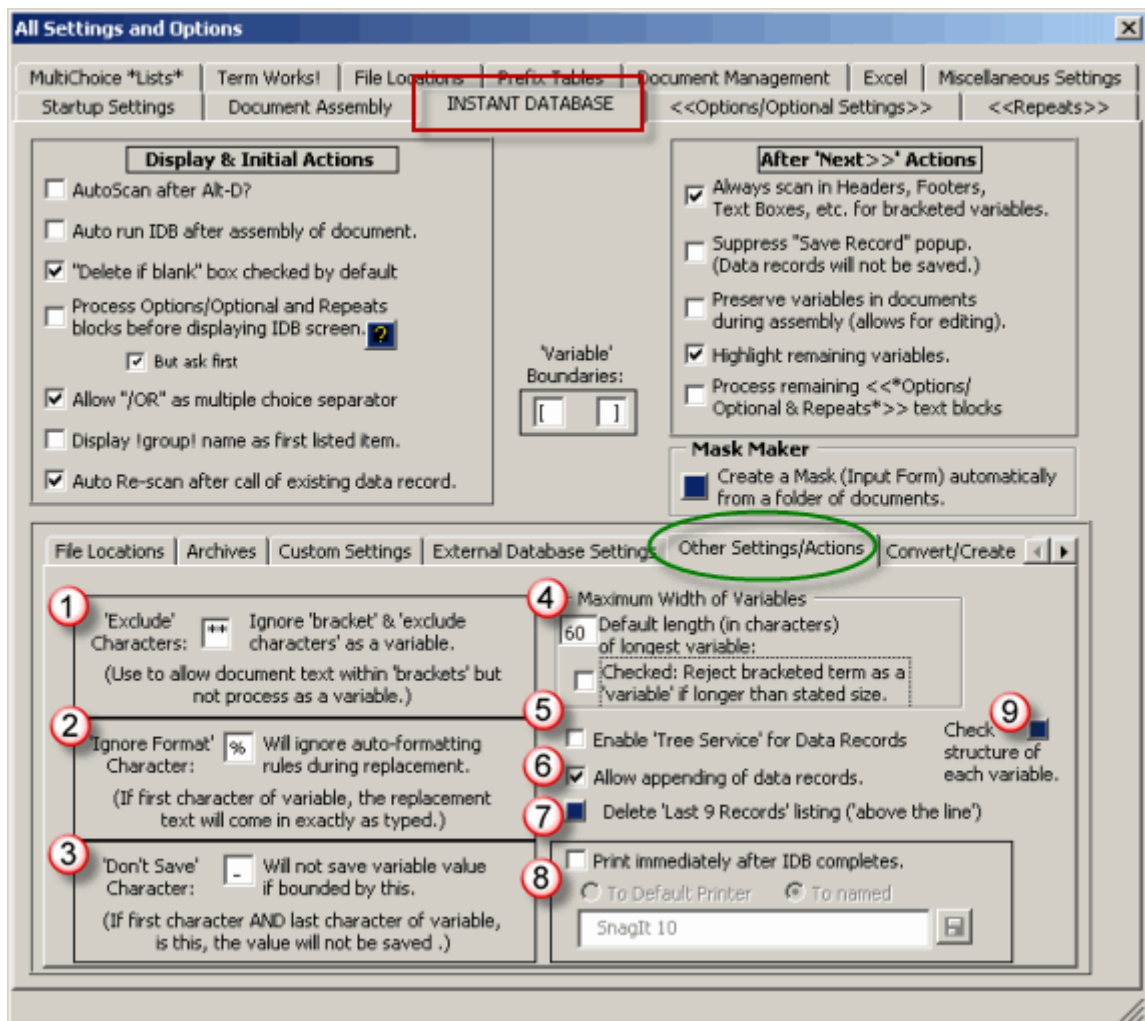
See Also: [Multi-Choice Lists](#) ⁴⁵⁸

[Linking to External Databases](#) ²⁸⁶

[Archiving IDB Records](#) ²⁵⁴

6.18 Still More IDB Settings

As stated in the previous screen, the settings for Instant Database processing is practically infinite. Here are even more that are available behind the "Other Settings/Actions" screen:



1. You can tell Pathagoras to ignore as a variable any bracketed text that begins with a designated character set.

2. You can tell Pathagoras to replace a variable exactly as typed. (Use when you don't want Pathagoras to covert a variable to ALL CAPS or all lower case when the variable within the text is written as such.)
3. A variable written [-Date-] or [-Name of Document-] will be replaced with the designated replacement, but the replacement will not be saved with the other terms in the record. Look at this as a 'temporary' variable.
4. Designate the default maximum width which Pathagoras will 'automatically see' a bracketed set of word as a variable. If longer, Pathagoras will Ask if you want to process it as a variable. Check the box and any bracketed text longer than 60 (or whatever) will be ignored.
5. Check to store your data records in sub-folders.
6. Pathagoras allows you to append data records. So you can select one record, append a set of related data to that record and end up with an augmented 'third' data set. An advanced function, but it allows you to have a 'header' record of standard client data, and then any number of secondary records for specific matters. When used, the augmented record can be saved, but is it probably best if it is not. (No need to re-save -- and possibly have to edit -- duplicated information.
7. Delete the 'above the line' listing of records, which typically show the last 9 recalled data records. (Only the listing is deleted, not the records.
8. If you want to immediately proceed to printing after the Instant Database record is processed, check this box.

6.19 Other Uses for IDB

Simple Find and Replace using IDB

Display the Instant Database Form.

Type into the left hand column any text that you want replaced throughout the document. While the terms could (and typically will be) variables in the current document, the left-column terms could just as easily be standard words that you simply wish to be replaced with other words.



The Instant Database module is really nothing more than an elaborate “Find & Replace” tool. However, unlike Find & Replace where replacements can be made one at a time, Pathagoras can make replacements 10, 20, 100, etc., at a time.

- In the right column, type the word or phrase that you want to replace what you have typed at the left.
- When all “find” and “replace with” terms are listed, press the Next>> button.
- The questions “Do you want to save the data as a personal record?” will next appear. Unless you want to save the data on the screen for reuse (unlikely in this scenario), answer “No”.
- Very quickly, Pathagoras will locate each instance of the “find” text and replace it with the corresponding “replace with” text.

Make a 'Sex Change' Mask

This is not what you think. This is simply a different application of Pathagoras' Instant Database by which you can reverse 'he' and 'she' and 'him' and 'her' throughout an entire document. The following 'pattern' of left and right column entries will take care of the issue.

This example also illustrates how the IDB mask can be programmed to handle simple housekeeping chores when you do not want to create an elaborate mask. (Notes: there must be spaces on both ends of each word. Doing so prevents the possibility of the change from 'he' to 'she' from affecting words that contain 'he'. So 'the' (which contains 'he') does not become 'tshe'. The 'xxx' suffix serves to make sure that the order of replacements are maintained.

Variable	Replacement Text
----------	------------------

she	shexxx
-----	--------

her	[himxxx/hisxxx]
-----	-----------------

hers	hisxxx
------	--------

female	malexxx
--------	---------

woman	manxxx
-------	--------

(add any other female sex based term)

he	she
----	-----

him	her
-----	-----

his	[her/hers]
-----	------------

male	female
------	--------

man	woman
-----	-------

hexxx	he
-------	----

hisxxx	his
--------	-----

malexxx	male
---------	------

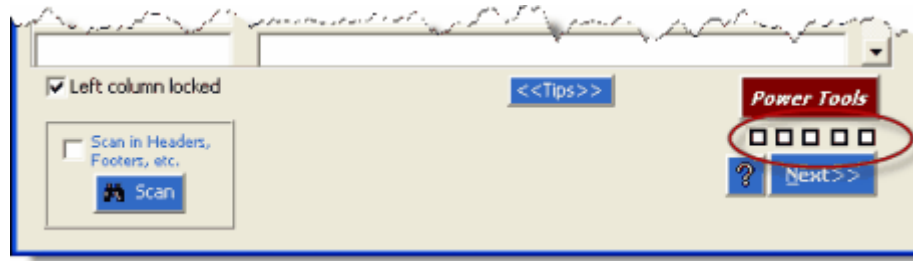
manxxx	man
--------	-----

➔ This is but one of any number of 'transformational' masks that you could create. (Something like the above is not necessary with a typical 'Pathagorized' document.

[Groupings](#)¹⁵¹, discussed on an earlier page can easily take care of gender based pronouns. But the above, and others like it, are 'cool tricks' when it comes to modifying documents that have not yet been Pathagorized.)

6.20 Advanced Array

Immediately above the Next button are 5 squares that (intentionally) appear to be ornamental. However, each of these square serves a purpose. We call this section of the screen the 'Advanced Array' because the boxes of the array should be used only by advanced users -- those who are already familiar with basic program operations.



Advanced Array

From left to right, here is what each box does when checked:

- ☑ **Apply the Instant Database against all open documents.** This is a wonderful tool if you typically compose several documents for a client or customer at the same time, have them displayed in separate editing screens and want to complete them all at the same time.



This feature can also be used to Scan all open documents for variables. Just check it before you click the Scan button.



This feature can also be used as multi-document '*Bulk Edit*' tool. By 'Bulk Edit' we mean searching and replacing multiple terms in multiple documents.

Keep in mind that IDB is essentially little more than a fancy "search and replace" utility. **It can be used to modify any of document.** So, from a blank IDB screen, type the 'old text' in the left column and type the new (replacement) text in the right column. Click the box and then press Next.



The resulting action is a standard 'search & replace', but it will be against *all open documents*.

- ☑ **Do not process <<*Options/Optional*>> text blocks** after processing IDB screen.
- ☑ **Do not attempt to preserve emphasis** (bold, underlines, CAPS, etc. when processing replacements)
- ☑ **Suppress "Do you want to save/update the matter record?" popup.** The question will not be asked, and changes to existing Matter record will not be preserved. (You can permanently suppress the popup via the Instant Database setting screen. The switch is under the 'Miscellaneous Settings' tab.)
- ☑ **Turn off screen display.** (Speeds up processing of very large documents.)

i Don't worry if you cannot remember the Advanced Array functions. Simply hover the mouse over any box in the advanced array. A prompt will appear describing the function of the box.

6.21 Screen Shots Depicting Screen Features

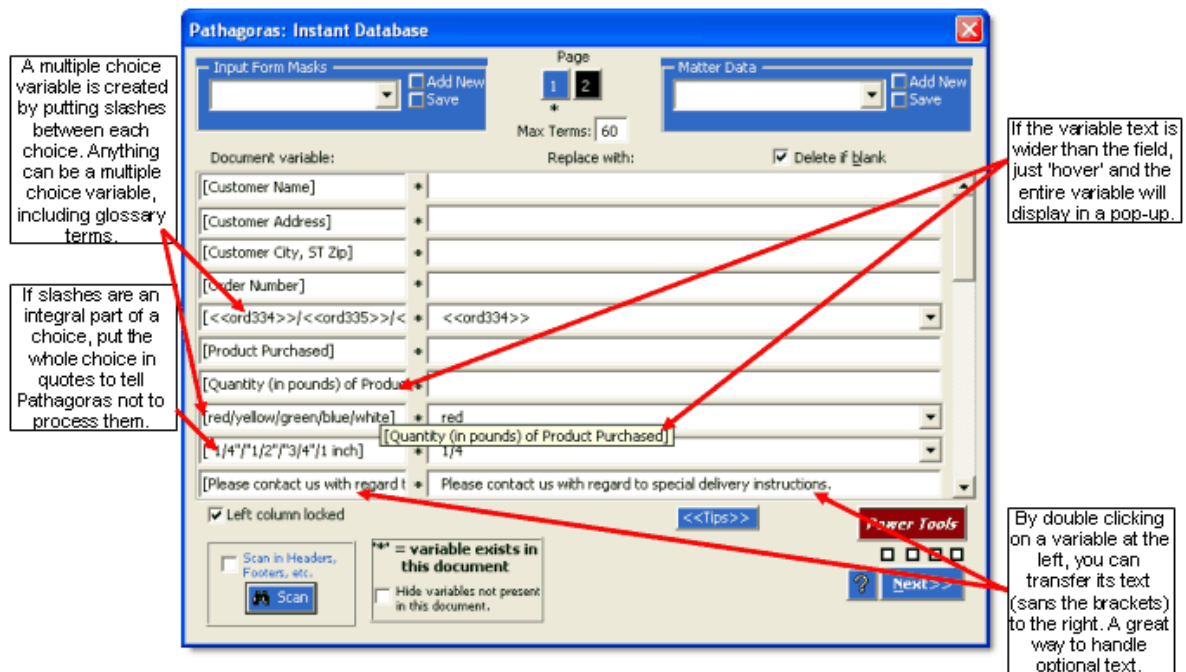


Figure 1. Results from a <Scan>.

All variable displayed in the left column are plain text, [bracketed variables] in the underlying document.

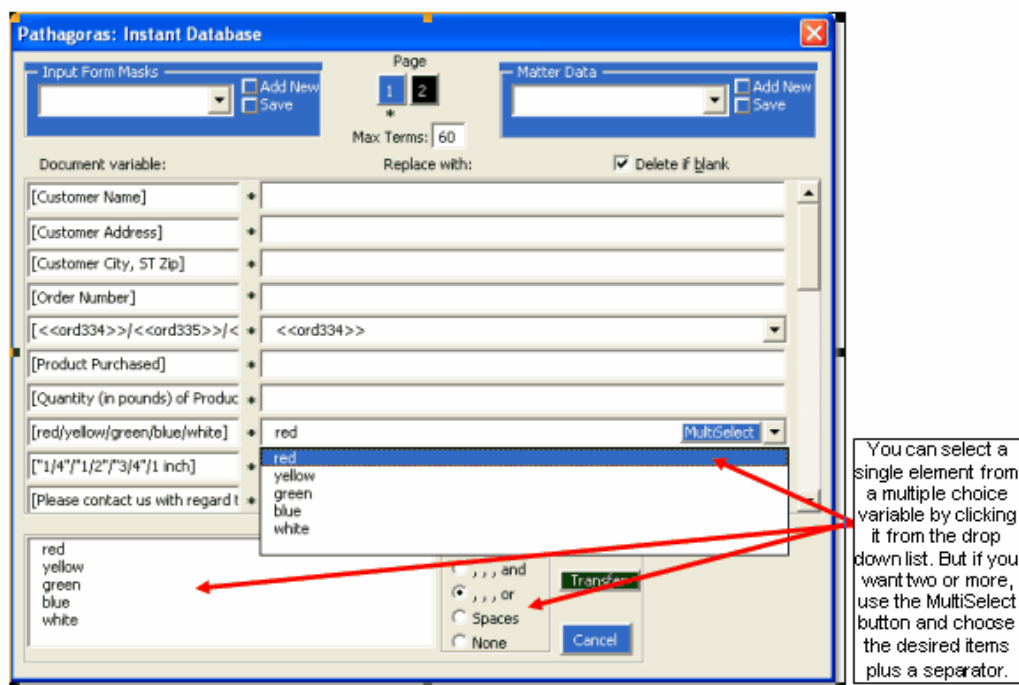


Figure 2. IDB: Multiple Choice Selector [203]

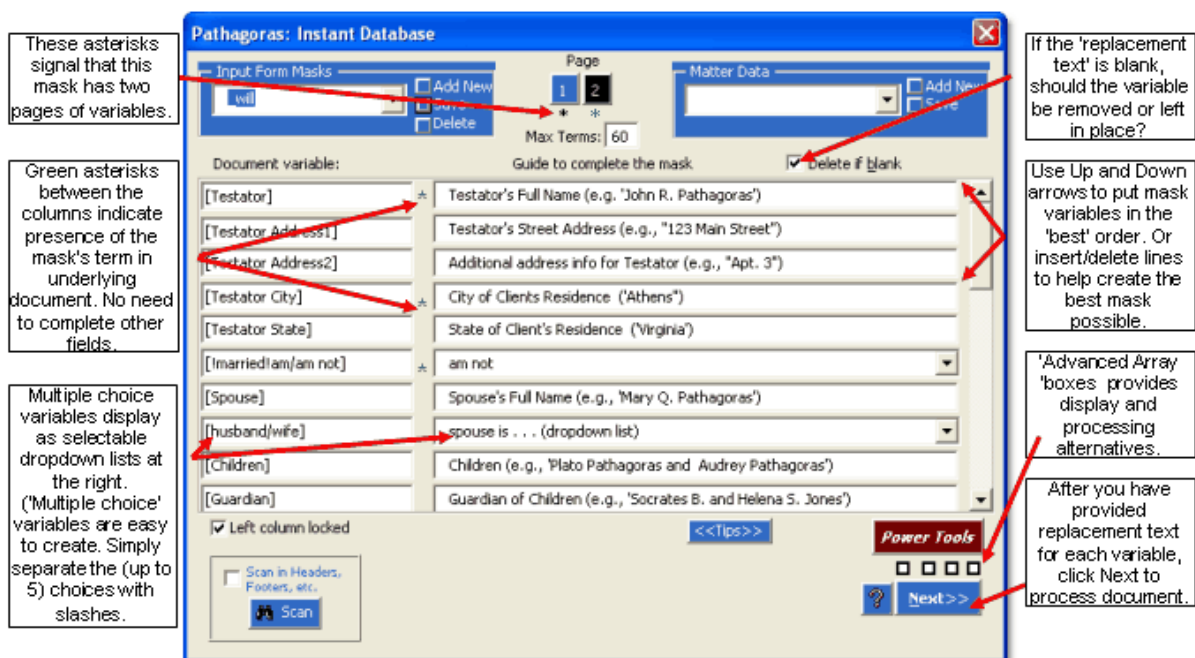


Figure 3. Instant Database Mask [260] w/ Completion Tips (ready to be saved as a mask).

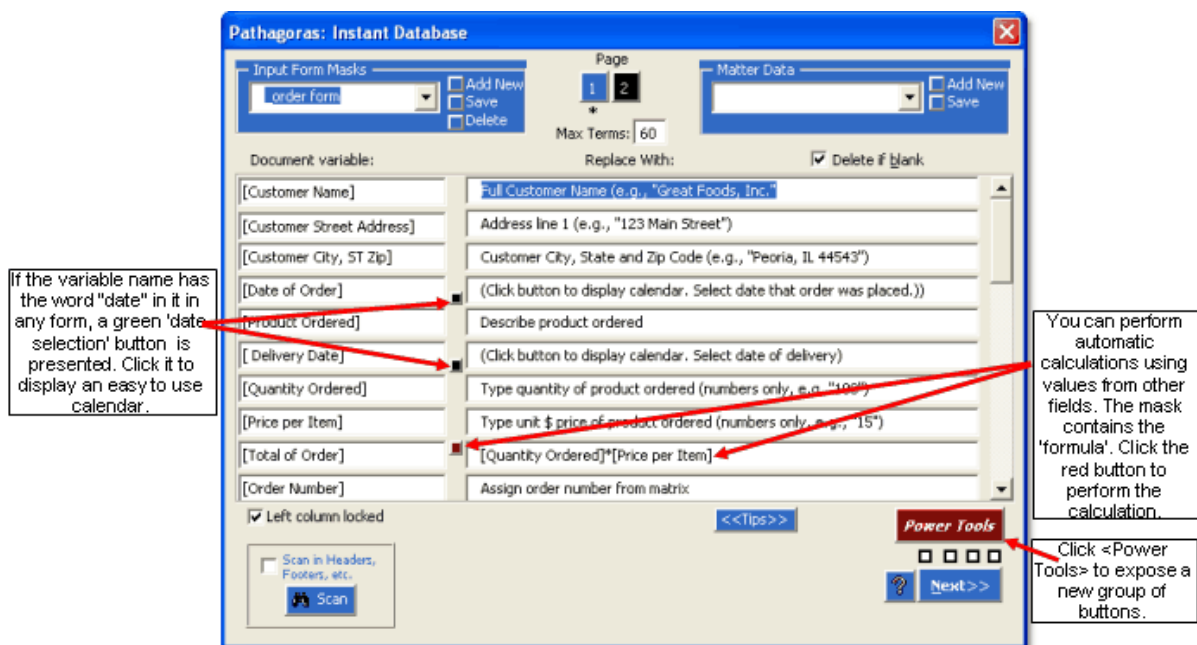


Figure 4. More IDB features, including [Date Picker](#)^[224] and [Calculator](#)^[199] buttons on display.

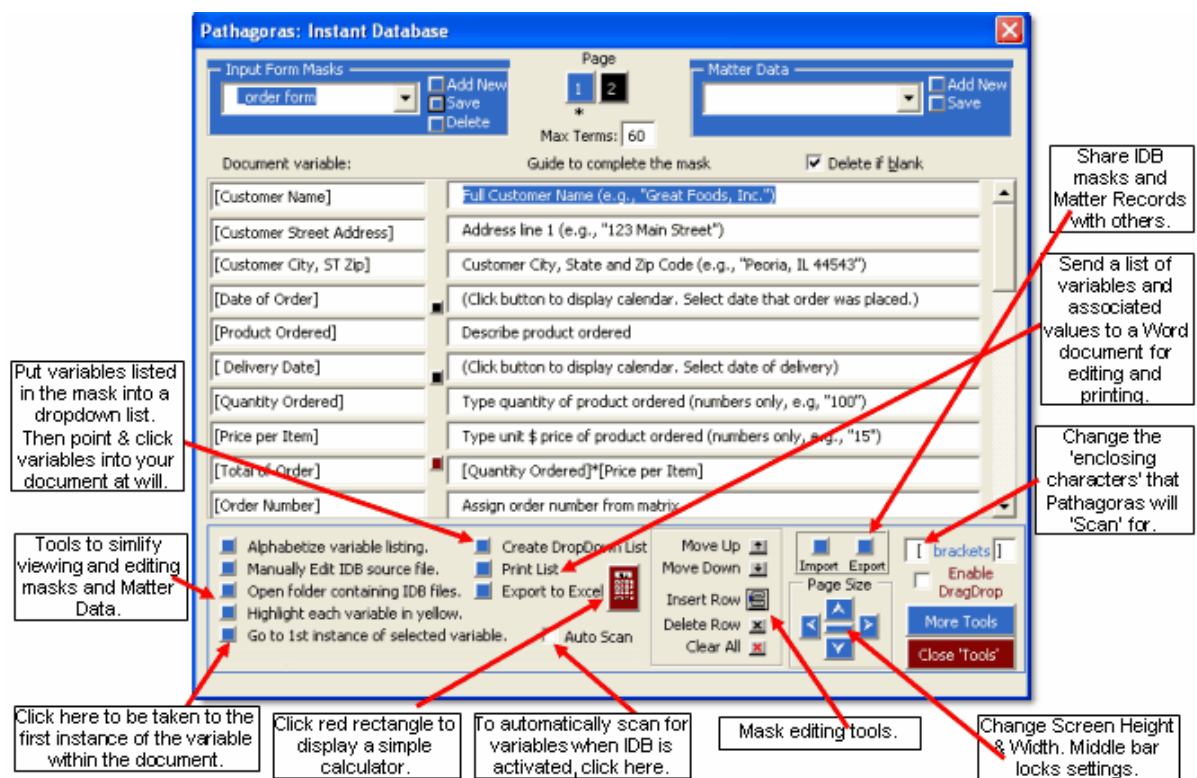


Figure 5. The additional IDB features show under '[Power Tools](#)'^[210].

The Pathagoras System

Instant Database Functions

Part



VII

7 Instant Database Functions

Pathagoras offers a wide variety of tools and shortcuts which allow you to easily conform your documents to designated office standards. They include:

- date formatting,
- date math,
- navigation tools which allow you to set a 'variable' to a folder. Use, for example, when you want to insert an entire document (such as a property description) in place of the variable.
- pairing of multiple variables to the same value (for situations where 'Client' and 'Plaintiff' are the same person for one case, but 'Client' = 'Defendant' in another)
- spell out functions: dollar and other number values; percents; fractions

These functions are discussed in the below sections of this Manual.

7.1 Calendar and Date Math Features

Instant Database: Calendar and Date Math features

If Pathagoras detects a variable which includes the word “date” in any form, Pathagoras will display a small green button that will appear between the two columns on the same row as the variable being processed.

When clicked (optional), Pathagoras will present a calendar from which you can

- (1) pick a date and
- (2) perform date math (add a certain number of days, weeks, months or years to a starting date).

Examples of "date" words that will trigger the display of the calendar button: 'Date', 'Birthdate', 'Date of Sale', 'Intimidate' (sorry).

Screen Shots:

Pathagoras: Instant Database

Page 1 2

Input Form Masks: Real Estate [Add New] [Save] [Delete]

Matter Data [Add New] [Save]

Max Terms: 60

Document variable:

[Landlord]	Name of our client
[Tenant]	Name of tenant
[Address of Property]	full street address
[Inception Date]	beginning date of lease
[Termination Date]	end date of lease
[Contract Date]	date lease signed
[monthly rent]	monthly rental amount
[deposit]	deposit (typically)

Left column locked

Scan in Headers, Footers, etc. [Scan]

Hide variables not present in this document.

Show Tips

Power Tools

[?] Next>>

[Inception Date]

Today September 2008

Sun	Mon	Tue	We	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

September 28, 2008

OK

Whenever a variable contains the word 'date' (in any form), a small green button between the columns will display. Click it to reveal the calendar.

Select a date. (Insert the selected date into the Instant Database screen by 'double clicking' or pressing the okay button.)

To perform date math, click the Calculator to the right of the calendar overlay.

The screenshot shows the 'Pathagoras: Instant Database' application window. A modal dialog box titled '[Inception Date]' is open, displaying a calendar for September 2008. The date 'September 28, 2008' is selected. To the right of the calendar is a 'Date Math' section with a text input showing 'September 28, 2008', a numeric input set to '1', and radio buttons for 'Days', 'Months', 'Weeks', and 'Years'. Below these is a 'Calculate' button and an 'Expiry Date' checkbox. The result of the calculation, 'October 27, 2008', is shown in a text box, with a 'Transfer to IDB' button below it. The main application window in the background has a top bar with 'Input Form Masks' (set to 'Real Estate'), 'Page' (1 of 2), and 'Matter Data' fields. Below this is a list of document variables: [Landlord], [Tenant], [Address of Property], [Inception Date], [Termination Date], [Contract Date], [monthly rent], and [deposit]. At the bottom, there are checkboxes for 'Left column locked', 'Scan in Headers, Footers, etc.', and 'Hide variables not present in this document', along with a 'Show Tips' button and a 'Power Tools' button.

Perform date math by adding or subtracting days, weeks, months or years.

The option 'Expiry date' is one period minus one day to reflect (as do most leases) the last day of the term.

Notes:

- If you have created a mask, you can perform simple date-addition directly on face of the the Instant Database screen. See the [next section](#)¹⁹⁹ for the steps to set up the formula.
- The format for the return value of the date (e.g., "01/01/07", "January 1, 2007" etc.) can be set via the Instant Database Settings Screen. [See this link](#)²¹².
- To write out the date in an extended fashion (e.g., "1st day of January, 2015"), you can take advantage of a setup shortcut. Simply add the text "(ext)" after the variable name, but before the closing bracket.
- The variable [Date of Closing] and [Date of Closing (ext)] are functionally the same variable. Both return the same 'date.' However, with the bullet point immediately above in mind, the former will return, for example, "January 1, 2015" whereas the latter will return "1st day of January, 2015". You need only provide the date in one location. Pathagoras completes the correspondent cell.

7.2 In-line Math and Date Math

As the previous sections indicate, Pathagoras has the ability to perform simple mathematical calculations and concatenations of 'text'. While pretty 'good' for a plain-text based program, it still required that the user create a mask to store the formula, making the documents a little less universal.

Masks are actually wonderful things, but the reality is that they are more challenging to create (or to remember to create).

Beginning with version 2012.2, Pathagoras offers the user to ability to insert math and concatenation formulas 'in-line' with the source text (as opposed to requiring the formula to be saved in a mask).

A precise structure is required (examples shown below), but its really not that hard:

The 'formula' is:

1. square brackets surrounding the entire variable term;
2. the call-to-action "#Math#" (no quotes) placed immediately after the opening bracket;
3. the variable name;
4. a equal sign to indicate the equivalency;
5. the formula, which is simply a reference to the existing variable, and the math you wish to perform. Note that there will always be a nesting of the variables, and the closing of the variable will very frequently be a double ']]'.

Here are some examples. The first elements are simply plain variables. They need to be completed in order for any in-line math to be performed. There is no order requirement EXCEPT that all references must be on the same Instant Database page as the formula.

Simple Math ('total price', 'half unit price' and 'half total price' are the target variables):

```
[Quantity]
[Price]
[#Math#total price=[Quantity]x[Price]]
[#Math#half unit price=[Price]/2]
[#Math#half total price=[Quantity]x[Price]/2]
(Note: multiplication must be indicated by 'x', not '*')
```

Date Math ('termination date' is the target variable):

```
This contract will begin on [start date].
It will automatically renew on [#Math#termination date=[start date] + [3 months]], and every
3 months thereafter.
```


If you want to return the 'expiry' date (the day before the 'typical' return date, use this setup:


```
This contract will begin on [start date].
It will expire at midnight on [#Math#termination date=[start date] + [3 months(exp)]].
```


Age Math ('Age of H' is the target variable):

```
Husband was born on [Birthday of H].
He is [#Math#Age of H=[Today]-[Birthday of H]] years old.
```

(Note: When the variable [Today] is in the formula, you need not pre-set its value. Pathagoras will assign the current date to the variable.

 One might say that the above formulas are pretty complex to set up. While it may be true, keep this in mind: Math in any program requires setting up a formula. With Pathagoras (as opposed to other programs), the formula is shown 'in-line' and on the face of the document. That makes it more accessible for editing, and more accessible for 'seeing' what is happening.

 Once the variable in the in-line formula has been 'calculated,' the variable can be used in other parts of the document without the 'formula'. So once 'total price' or 'half unit price' or 'termination date', etc. have been used and processed in a document, that variable can be referenced later in the document as a 'normal' variable, e.g., [total price] or [half unit price] or [termination date]. After all, it's just a plain text variable!

 By default, Pathagoras will return the calculated value in number of days. To return weeks, months or years, be sure to include the term 'weeks' 'months' or 'years' somewhere in the formula. E.g.,

John Smith worked for Spacely Sprockets for [#Math#duration in weeks=[Client 1 Empl Start Date] -[Client 1 Empl End Date]] weeks.

7.3 <<Folder>> Function

As discussed in the immediately preceding section, not only can you provide a hard coded reference to file (or a multiple choice list of file name) you can provide a reference to an entire folder of documents. Chose from among the current contents of that folder.

How to:

After you have pressed Alt-D (or otherwise displayed the Instant Database screen), complete the variables as you ordinarily would. When you encounter a variable that you wish to be replaced with an entire document, here are the steps:

Manually (type by hand):

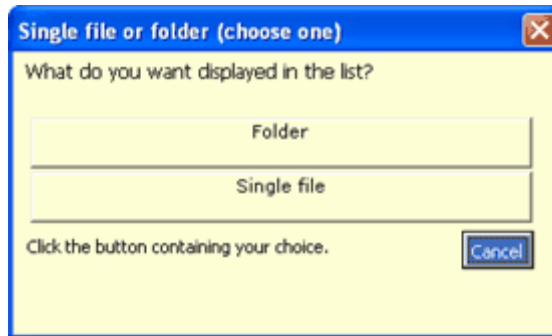
If you know the full name of the document you want to be inserted in place of the variable, you can simply manually type it into the field. You must enclose the name within "<<" and ">>" markers (This is not 'code.' These are simply doubled 'lesser than' and 'greater than' characters found above the comma and period on your keyboard.).

For example:

```
<<N:\data\property descriptions\123 Main Street.doc>>
```

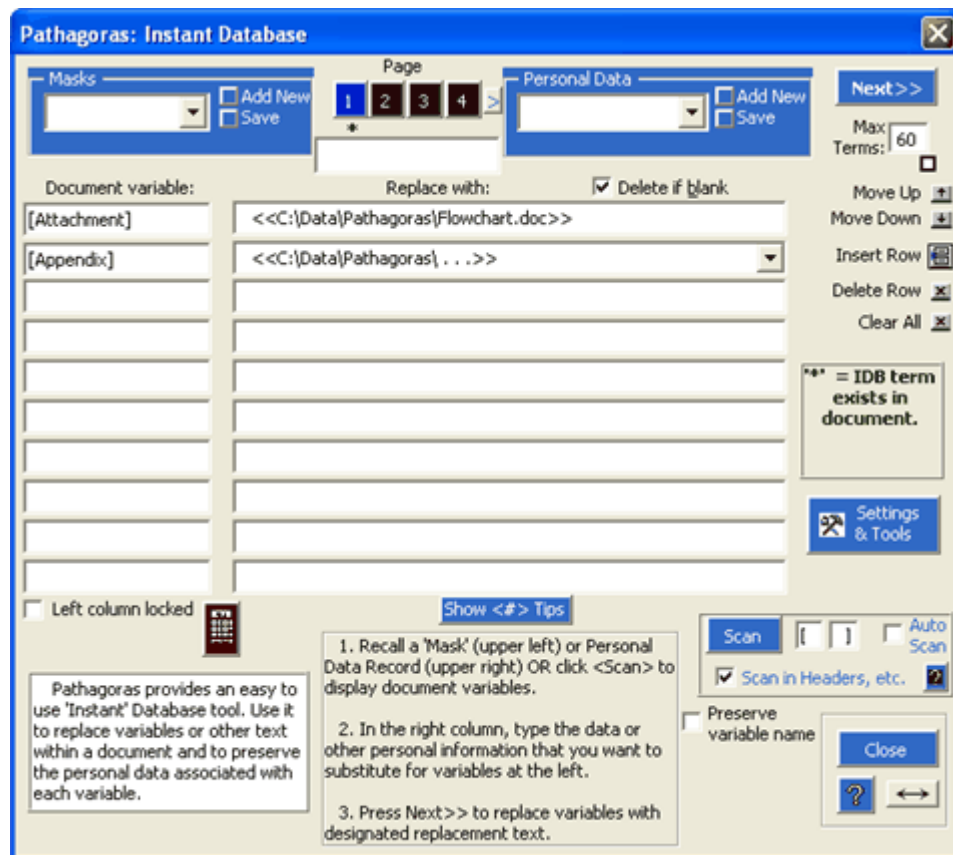
Navigate to file or folder:

You can also navigate to the file, or to the folder containing a number of possible file choices. To start this more automated process, *Shift-click* inside of the 'value' field to the right of the variable. When you Shift-click, a menu like the below will appear:



Choose the appropriate selection. "Folder" if you want Pathagoras to display all files in the target folder or "Single file" if you want to select a specific file. Navigate to the file or folder (if 'folder,' navigate inside of the folder and select a single file, it matters not which one --its just to lock in the folder selection. Press OK.

The file or the folder will appear in the IDB screen. If you selected "Folder" all of the files in the target folder are shown in a drop down list.



The 'assign variable to folder' feature is also useful when want to save the Instant Database data as a Mask or Data Record, but don't always know the name of the file you want to use as you are creating the mask. Not a problem. Link the variable to a folder and, at document

assembly time, select the document from the dropdown list of the folder's content. (Yes, the desired document will at least have to be within the designated folder, but hopefully that will not create too much of a burden.)



"This seems a lot like a DropDown List. How does it differ?"

Well, it actually is based in the DropDown List module. Except for its location, they are indistinguishable in their ultimate operation. The differences are, however, real, if not subtle:

- With DropDown Lists, you insert the selected document or term where the cursor is at the time you make the call from the list. To make sure it is in the right place, you may have to navigate to the right location in your document before you select the text you want to insert. With the Variable=Folder feature described above, the selected document or term is inserted at the location of the variable. you don't have to worry about finding it because it is already set.
- With DDL, the selection is always optional. You insert a clause because you see a need, but there is no reminder. With the Instant Database version, the selection is obligatory. There is an actual variable in a document that needs to be completed, and there is an affirmative reminder to you of that need.
- Plus (and this is a *big plus*), if the content of the target file is needed in multiple locations in the document under construction (e.g., property descriptions are frequently found in multiple locations), Pathagoras will automatically replace all instances of the variable with the target file when the Instant Database routine is run.

7.4 '=' Equivalency Function (single element)

A user wrote:

"We use the variable '[OurClient]' to produce documents and letters to or concerning our client. But for documents prepared in the course of litigation, 'Our Client' is sometimes the '[Plaintiff]' and sometimes the '[Defendant]'. But our client is always '[OurClient]'. It sure would be nice to have a way to quickly equate the data recorded next to [OurClient] to either the [Plaintiff] or the [Defendant] value field in the IDB mask without risking a typo."

Pathagoras can handle that. You can set the value of one variable to automatically be equal to the value of another variable. Simply put in the following equivalency function:

"=[variablename]" (no quotes) in the right column of the IDB screen next to any other variable. See the below screen shot where the variable [Plaintiff] is equal to the value of [ClientName].

The variable[ClientName] found in position #1 of the variable list was inserted in the third line, with an '=' sign added to create the equivalency.

When Pathagoras detects the '=' sign adjacent to a bracketed variable at the 'right', it displays a red button between the two columns.

When you click on the button, "=Client Name" will become "John Q. Doe".

If you click the red button again, the 'equation' is restored. That is in case you need to edit the formula or (if a multiple choice selection is supplied) choose another item from the list.

You can concatenate two, three or four individual variables into a new variable. E.g., [Full Name] = [First Name] & [Middle Name] & [Last Name]. Click this link for more information

See [next page](#) ²³¹ for a discussion on how to assign a role to one of several possible actors.

7.5 '=' Equivalency Function (*Actors* and their Roles)

On the [previous page](#) ²³⁰, we assigned the value of a particular variable in a mask to that of a known precedent variable. In that example, we assigned [Plaintiff] in the document to be equal to [Client Name] in the database. We call this an A = B equivalency.

But oftentimes the direct A = B assignment is not practical. When you are drafting a Will, for

example, the Executor can be one of several people. It most often will be the spouse, but it can just as easily be a child or another relative or even a non-relative. And the possible actor filling the role of the Alternative PR can be quite large.

Pathagoras allows you to create equivalency lists that will allow you to select on-the-fly a desired 'actor' for each 'role' from a multiple choice selection.

Here is a sample 'multiple choice' equivalency:

[=[Spouse Name]/=[Child@1 Name]/=[Child@2 Name]/=[Child@3 Name]]

The variable

<p style="text-align: center;">LAST WILL AND TESTAMENT</p> <p style="text-align: center;">OF</p> <p style="text-align: center;">[TESTATOR NAME]</p> <p>I, [TESTATOR NAME] declare this writing to be my Last Will and Testament.</p> <p style="text-align: center;">Family</p> <p>I am married and my spouse's name is [Spouse Name].</p> <p>We have two children together, namely, [Joint Child@1 Name] and [Joint Child@2 Name]. I have no other children from prior relationships</p> <p style="text-align: center;">Personal Representative</p> <p>I hereby name as my Personal Representative [PR:=[Spouse Name]/=[Joint Child@1 Name]/=[Joint Child@2 Name]] and if my said Personal Representative cannot server, I appoint [AltPR:=[Spouse Name]/=[Joint Child@1 Name]/=[Joint Child@2 Name]] in his or her place.</p>

Because the equivalency is used twice in the document, we assigned titles 'PR:' and 'AltPR:' to the respective appearances of the formula. That way, we can distinguish one use from the other.
(We could also use a !groupname! in lieu of the title.)

Copy and paste the above text into a document. Press Alt-D (Instant Database) to display the Instant Database screen and the scan the document. Provide values for the actors (Testator, Spouse and Children). Example:

Pathagoras: Instant Database

Input Form Masks: ☐ Add New ☐ Save

Page: 1 2

Existing Records: ☐ Add New ☐ Save

Max Terms: 60

Document variable: Replace with: ☐ Delete if blank

[TESTATOR NAME]	* John Q. Doe
[Spouse Name]	* Mary R. Doe
[Joint Child@1]	* Franklin J. Doe
[Joint Child@2]	* Margaret M. Doe
[PR: *Actors*]	* PR: <input type="text"/>
[AltPR: *Actors*]	* AltPR: <input type="text"/>

☒ Left column locked

☒ Scan in Headers, Footers, etc.

* = variable exists in this document

☐ Hide variables not present in this document.

☐ & hide completed variables.

<Shortcuts> Power Tools

? Next>>

Now its time to assign an actor to the role of PR and an actor to the role of AltPR. To do so, just click the dropdown list to the right of the appropriate row. Note a red button appears between the two columns.

Pathagoras: Instant Database

Input Form Masks:

Page: 1 2

Max Terms: 60

Existing Records:

Document variable: Replace with: ☐ Delete if blank

[TESTATOR NAME]	+ John Q. Doe
[Spouse Name]	+ Mary R. Doe
[Joint Child@1]	+ Franklin J. Doe
[Joint Child@2]	+ Margaret M. Doe
[PR: *Actors*]	= [Spouse name]
[ALTPR: *Actors*]	= [Joint Child@2]

☒ Left column locked

☒ Scan in Headers, Footers, etc.

* = variable exists in this document

☐ Hide variables not present in this document.

☐ & hide completed variables.

<Shortcuts> **Power Tools**

Click the button and Pathagoras returns the appropriate value to the appropriate line.

Pathagoras: Instant Database

Input Form Masks:

Page: 1 2

Max Terms: 60

Existing Records:

Document variable: Replace with: ☐ Delete if blank

[TESTATOR NAME]	+ John Q. Doe
[Spouse Name]	+ Mary R. Doe
[Joint Child@1]	+ Franklin J. Doe
[Joint Child@2]	+ Margaret M. Doe
[PR: *Actors*]	+ Mary R. Doe
[ALTPR: *Actors*]	+ Margaret M. Doe

☒ Left column locked

☒ Scan in Headers, Footers, etc.

* = variable exists in this document

☐ Hide variables not present in this document.

☐ & hide completed variables.

<Shortcuts> **Power Tools**

Assigning Equivalencies to a MultiChoice *List*:

The equivalency formula "[Spouse Name]/=[Child@1 Name]/=[Child@2 Name]/=[Child@3 Name]" is not too bad to work with. It is short and doesn't take up much room in the document. (But, for the novice, it still may be hard to look at.)

But what if things get more elaborate (i.e., longer). Instead of 3 or 4 possible actors per role, perhaps you have 15 or 20 (or more). Here is a possible list of actors who can fill various roles in estate planning documents.

Client, Spouse, JointChild@1, JointChild@2, JointChild@3, JointChild@4, SepChild@1, SepChild@1, SepChild@1, SpouseChild@1, SpouseChild@2, SpouseChild@3, Mother, Father, Sibling1, Sibling2, Sibling3, Sibling4, Beneficiary1, Beneficiary2, Beneficiary3

I think you get the point. It's just not practical to clutter up the document with all the choices each time want to assign specific role (Settlor, Trustee, PR, AltPR, Guardian, AltGuardian, etc.) to a previously named actor.

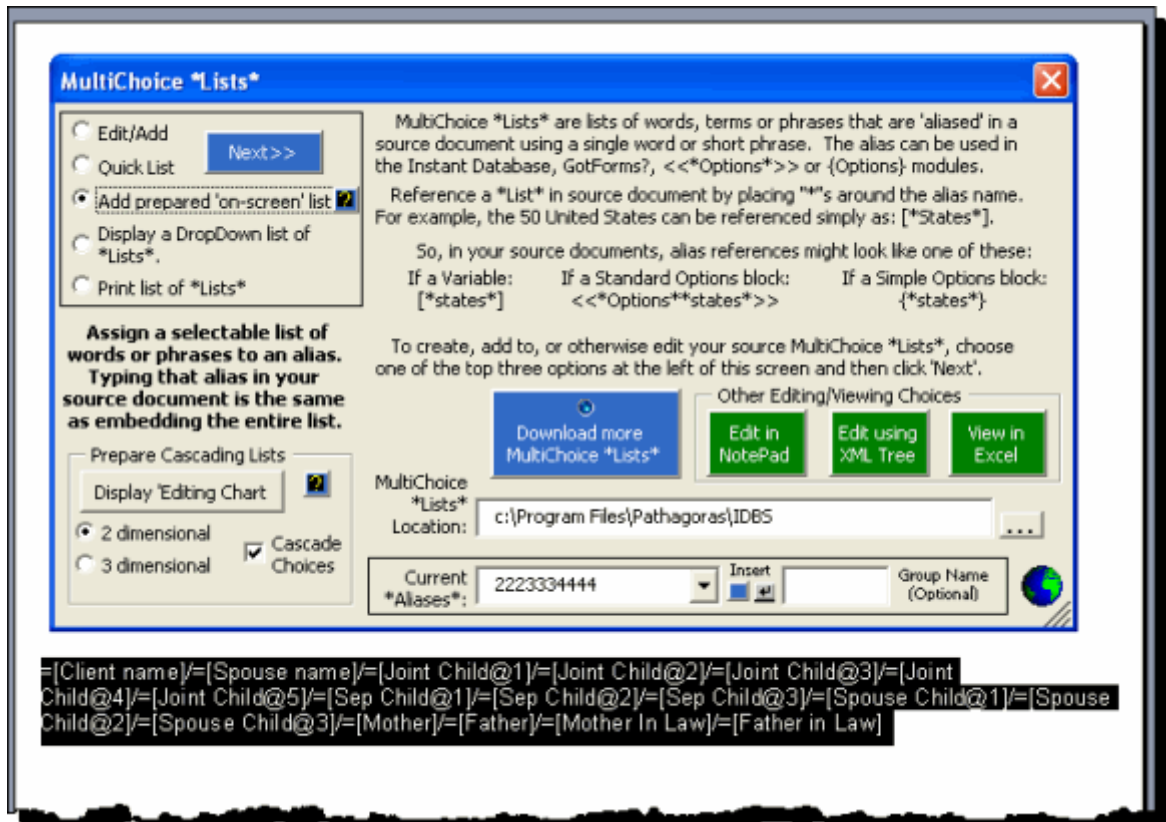
MultiChoice *Lists* handle this quite elegantly. You may recall that you can assign a word or term (an 'alias') to represent all 50 United States, or the 212 countries of the world, or the 278 Ben & Jerry's Ice Cream flavors. So too can you represent the large variety of equivalents you wish to make available.

Pathagoras strongly encourages you to assign the various 'actors' to a MultiChoice *List*. The *List* could store all possible equivalencies. That way, you can easily assign an actor to a "role" just by pointing-and-clicking. And if an actor served dual or triple roles. Not a problem. Click here to read more about [MultiChoice *Lists*](#)⁴⁵⁸.

Example: Copy this list of potential actors to your editing screen:

= [Client name]/=[Spouse name]/=[Joint Child@1]/=[Joint Child@2]/=[Joint Child@3]/
 =[Joint Child@4]/=[Joint Child@5]/=[Sep Child@1]/=[Sep Child@2]/=[Sep Child@3]/
 =[Spouse Child@1]/=[Spouse Child@2]/=[Spouse Child@3]/=[Mother]/=[Father]/
 =[Mother-in-Law]/=[Father-in-Law]

Highlight the text. Call up the MultiChoice *Lists* editor from Pathagoras Features | Editing Tools. Click the 'Add Prepared On-Screen List' radio button. Your screen should look like this:



Click the next button. When you are asked for the 'alias,' you can use any term you want, but we suggest something like 'Actors'.

LAST WILL AND TESTAMENT

OF

[TESTATOR NAME]

I, [TESTATOR NAME] declare this writing to be my Last Will and Testament.

Family

I am married and my spouse's name is [Spouse Name].

We have two children together, namely, [Joint Child@1 Name] and [Joint Child@2 Name]. I have no children from prior relationships.

Personal Representative

I hereby name as my Personal Representative [PR:*Actors*] and if my said Personal Representative cannot server, I appoint [AltPR:*Actors*] in his or her place.

Copy the above text into a new document. Follow the identical steps as in the example at the top of the page.

Note: it doesn't matter that not all variables exist in the document, or that not all actors can

fill the designated role. (The Client cannot possibly be his own Personal Representative.) What is being demonstrated is the ease and elegance by which roles can be filled without having to retype names.

7.6 Concatenation of Variables

Concatenation:

You can combine ('concatenate') two, three or four pre-existing variables to create a new variable. Type (in the left column if the IDB screen) the new variable beneath the existing variables. (It does not have to be immediately beneath an existing variable. It can also be on a different page.) In the right column, type the two to four existing variables that you want to combine. You must include the '=' sign to indicate a 'calculation event' and the formula, and an ampersand ('&') to indicate the concatenation.

So, if you have the variables [First Name] and [Last Name] in your Instant Database mask, but wish to use the variable [FullName] in the document, you do not need to type the First and Last Name values again. Just provide in the Instant Database mask the target variable '[FullName]' at the left and the equivalency formula at the right.

Variable		Value
[First Name]		John
[Last Name]		Doe
[FullName]	■	=[First Name]& [Last Name]

A red square appears in the 'middle' column when the '=' is present.
Click the red square between the two columns and the equivalency is 'calculated'.

Variable		Value
[First Name]		John
[Last Name]		Doe
[FullName]	■	John Doe

Click it again and the formula is restored for editing, if necessary.

In-line concatenation:

The above can be also accomplished 'in-line' (i.e., within the document body. (This avoids the need for a mask.)

Pathagoras will see the in line formula as a complete variable. It will automatically create the equivalency formula at the right with no action on your part. Click the red button that will appear between the two columns to toggle between the formula and the calculated value.

Let's assume that [First Name] and [Last Name] (the 'precedent' variables) reside in the 'upper part' document. You now want to reference the concatenated value of those two variables somewhere below the first appearance of the precedent variables. Here is the formula

that would reside in the actual document.

[#Concat#Full Name=[First Name]& [Last Name]]

When the document is scanned, the precedent variables are (presumably) found first and inserted into the IDB screen. When the #Concat# instruction is encountered, entire block is place into the variables list (left side) the variable. The formula is then automatically parsed from the #Concat# block and placed in the 'values' column of the IDB screen

Variable		Value
[First Name]		
[Last Name]		
[#Concat#Full Name]	■	=[First Name]& [Last Name]

(Note: just like in the IDB screen, the example doesn't show the entire very long variable in the left column. If you hover your mouse over the variable, it will display in its entirety.)

A red square appears in the 'middle' column when the '=' is present.

Variable		Value
[First Name]		Eleanor
[Last Name]		Roosevelt
[#Concat#Full Name]	■	=[First Name]& [Last Name]

Click the red square between the two columns and the equivalency is 'calculated'.

Variable		Value
[First Name]		Eleanor
[Last Name]		Roosevelt
[#Concat#Full Name]	■	Eleanor Roosevelt

Click it again and the formula is restored for editing, if necessary.

7.7 <S>pell out Function

If you wish your replacement text to be, or include, the spelled out equivalent of the Arabic numeral, precede or follow the number with

- <S> (for just a conversion to the spelled out version of the number) or
- <SS> (for both short and spelled out versions of the number).

If <SS> is placed first, the spelled out amount is the 'primary' value, and the number appears in parentheses. If <S> is placed last, the number is the primary value, and the spelled-out value appears in parenthesis.

The results are shown below the example.

Examples:

Variable (in document and on left side of IDB Screen)	Replacement Text (right side of IDB Screen)
[Quantity]	65344<SS>
[Number Ordered]	<SS>65344

Results:

Thank you for your order of [quantity] widgets.

will become:

Thank you for your order of 65,344 (sixty five thousand, three hundred forty-four) widgets.



You can '*pre-place*' the <S> or <SS> markers in your source documents. (Indeed, you are more likely to do this than manually type the markers into your Instant Database screen.) Just place the markers to the immediate left or right of the actual variable in the source text.

Example:

Document text:


Thank you for your order of [quantity]<SS> widgets.

If, when you display the IDB screen and type in 65344 beside the variable [quantity], the text will become:

Thank you for your order of 65,344 (sixty five thousand, three hundred forty-four) widgets.

7.8 <\$>Currency Function

If you wish your replacement text to include a currency equivalent of the Arabic numeral, precede or follow the number with

- <\$> (for just formatting to a dollar value style) or
- <\$\$> (for both short and spelled out versions of the dollar amount).
- If <\$\$> is placed to the left of the number, the 'spelled-out' amount is the 'primary' value, and the formatted \$number appears in parentheses. If <\$\$> is placed at the end of the number, the formatted \$number is the primary value, and the 'spelled-out' amount appears in parenthesis.
-  Append 'UC' or 'lc' to direct Pathagoras to return UPPER CASE or lower case 'spelled out' value.
(E.g., [Total Cost]<<\$UC>> or [Total Cost]<<\$lc>>)

Examples:

Variable (in document and on left side of IDB Screen)	Replacement Text (right side of IDB Screen)
---	---

[Total Cost] 65344<\$\$>

[Price Quoted] <\$\$>65344

Results:

The total cost of the project will be [Total Cost].

will become:

The total cost of the project will be \$65,344.00 (Sixty five thousand, three hundred forty-four and 00/100 Dollars).

We are pleased to offer to perform the work for [Price Quoted].
(document text)

will become:

We are pleased to offer to perform the work for Sixty-five thousand, three hundred forty-four and 00/100 Dollars (\$65,344.00)



You can '*pre-place*' the <\$> or <\$\$> markers in your source documents. (Indeed, you are more likely to do this than manually type the markers into your Instant Database screen.) Just place the markers to the immediate left or right of the actual variable in the source text

Example:

Document text:

The total cost of the project will be [Total Cost]<\$\$>.

If, when you display the IDB screen and type in 65344 beside the variable [quantity], the text will become:

The total cost of the project will be \$65,344.00 (Sixty five thousand, three hundred forty-four and 00/100 Dollars).

/-----/

Document text:


We are pleased to offer to perform the work for <\$\$>[Price Quoted].

If, when you display the IDB screen and type in 65344 beside the variable [quantity], the text will become:

We are pleased to offer to perform the work for Sixty-five thousand, three hundred forty-four and 00/100 Dollars (\$65,344.00)

7.9 <%%> Percent Function

If you wish your replacement text to include a 'spelled out' equivalent of the Arabic numeral, precede or follow the number with

- <%%> (for both numeric and spelled out versions of the percentage amount).
- If <%%> is placed to the left of the number, the 'spelled-out' amount is the 'primary' value, and the formatted number appears in parentheses. If <%%> is placed at the end of the number, the spelled out number is the primary value, and the 'spelled-out' amount appears in parenthesis.
-  Append 'UC' or 'lc' to direct Pathagoras to return UPPER CASE or lower case 'spelled out' value.
(E.g., [Contingency Fee]<%% UC>> or [Contingency Fee]<%% lc>>)

Examples:

Variable (in document and on left side of IDB Screen)	Replacement Text (right side of IDB Screen)
[Contingency Fee]	33.33<%%>
[Fee]	<%%>40

Results:

Document text:

We agree to represent you in this matter for a fee of [Contingency Fee] of the total amount recovered

will become:

We agree to represent you in this matter for a fee of 33 and 1/3% (Thirty-three and 1/3%) of the total amount recovered.

Document text:

We are pleased to offer to perform the collections work for [Fee] of any amounts recovered.

will become:

We are pleased to offer to perform the collections work for forty percent (40%) of any amounts recovered.



You can '*pre-place*' the <%%> markers in your source documents. (Indeed, you are more likely to do this than manually type the markers into your Instant Database screen.) Just place the markers to the immediate left or right of the actual variable in the source text

Example:

Document text:

We agree to represent you in this matter for a fee of [Contingency]<%%> of the total amount recovered

If, when you display the IDB screen and type in 33.33 beside the variable [Contingency], the text will become:

We agree to represent you in this matter for a fee of 33 and 1/3% (Thirty-three and 1/3%) of the total amount recovered.

7.10 <F>ormat Function

If you wish the replacement text of a 'simple' numerical value to be 'formatted' in the final text with appropriate separators, precede or follow the number with '<F>' (the placement of the modifier is not critical for this function).

Example:

Variable (in document and on left side of IDB Screen)	Replacement Text (right side of IDB Screen)
[Quantity]	65344<F>

Thank you for your order of [quantity] widgets.

will become:

Thank you for your order of 65,344 widgets.



You can '*pre-place*' the <F> marker in your source documents. (Indeed, you are more likely to do this than manually type the markers into your Instant Database screen.) Just place the markers to the immediate left or right of the actual variable in the source text.

Example:

Document text:

Thank you for your order of [quantity]<F> widgets.

If, when you display the IDB screen and type in 65344 beside the variable [quantity], the text will become:

Thank you for your order of 65,344 widgets.

7.11 <Fr>action Function

If you wish your replacement text to include a 'spelled out' equivalent of a fraction, precede the number with

- <Fr>
- <FR> or

- <fr>

(the case of the text indicates the case of the replacement.)

Examples:

Variable (in document and on left side of IDB Screen)	Replacement Text (right side of IDB Screen)
[Contingency Fee]	<FR>1/3
[Distribution]	<fr>1/4

Results:

Document text:

We agree to represent you in this matter for a fee of [Contingency Fee] of the total amount recovered

will become:

We agree to represent you in this matter for a fee of ONE-THIRD (1/3) of the total amount recovered.

Document text:

John Doe shall be entitled to [distribution] of my estate.

will become:

John Doe shall be entitled to one-quarter (1/4) of my estate.



You can '**pre-place**' the <Fr> markers in your source documents. (Indeed, you are more likely to do this than manually type the markers into your Instant Database screen.) Just place the markers to the immediate left of the actual variable in the source text

Example:

Document text:

We agree to represent you in this matter for a fee of <FR>[Contingency] of the total amount recovered

If, when you display the IDB screen and type in 1/3 beside the variable [Contingency], the text will become:

We agree to represent you in this matter for a fee of ONE-THIRD (1/3) of the total amount recovered.

7.12 <<Filename>> Function

Sometimes the best replacement for a variable is not a name or a quantity or a color.

Sometimes the best replacement for a variable is a complete document.

Pathagoras can handle this need with elegance and remarkable ease.

In another section ([Clause Sets](#))¹⁰⁴, you learned that you can create 'clause-pointers' by enclosing the target term within a set of plain text doubled angle brackets (e.g., <<term to recall>>). The target clause or document will be called into the document during the document assembly session.

This feature carries over intact into the Instant Database function. Simply type the clause you want recalled inside of plain text “<<” & “>>” markers (no quotes) on the right (replacement text) side of the IDB screen. E.g., <<term to recall>>.

When you finish all of the variable replacement values and press **Next>>**, <<term to recall>> will replace the original variable. Then, <<term to recall>> is processed in the same fashion it would be had it been in the source text at the inception.

The search for the target clause during the assembly session follows the sequence referenced in [Order of Search](#)⁹⁵.

- Examples:

Variable	Replacement Text
[Name & Address]	<<LasrisR>>
[Signature Line]	<<rhlsig>>
[Property Desc]	<<C:\data\property descriptions\JonesR\description.doc>>

- This feature works with multiple choice variables as well. Place the various possible signature blocks into the source document as an actual [variable]. The bracketed text would be picked up during the Scan and the options will be presented in a dropdown list in the Replacement side of the IDB screen.

Example:

[<<rhlsig>>/<<jpwsig>>/<<kmvsig>>].

When encountered during a Scan, the choices will be displayed in a drop down list at the right side of the IDB screen. You select one of the items. Let's say it was <<jpwsig>>. The characters <<jpwsig>> will be placed into the document in lieu of the multiple choice variable. Pathagoras then will hunt down the value for the term "jpwsig" and place it into the document. Two steps in one!

- As noted in the examples, you can reference a complete document within the <<term to recall>>.' Simply set out in the right side of the IDB form next to the appropriate variable the fully qualified document name in between the brackets.

Example:

<<C:\data\property descriptions\JonesR\description.doc>>.

When the replacements are made, the referenced document will be inserted at the location of the variable.

- **i** A fully qualified document reference as a target for a <<clause-pointer>> seems a

bit much to type. However, if you strategically organize your folders and adopt a consistent naming style, and create an [Instant Database Mask](#)²⁶⁰ which contains the basic information, this can actually work quite elegantly.

Using the example above, if you put all property descriptions in the "C:\data\property descriptions" folder, and always called them "description.doc" the only non-constant value in the 'variable' is the client name. So, when the mask is recalled, only the name needs to be replaced in the *<<clause-pointer>>*. (E.g.) "SmithP" for "JonesR".

i Pathagoras lets you navigate to the file you want to place between the double-angle brackets. Shift-click in the replacement text box and follow the prompts to select the file.

- This feature works with [Multichoice *List*](#)⁴⁵⁸ variables as well.

Example:

Let's assume that you have created a Multichoice *List* variable called [*Signature*]. The *list* consists of

```
<<rhlsig>>/<<jpwsig>>/<<kmvsig>>/<<emlsig>>/<<jsasig>>/
<<mlssig>>" (etc.).
```

When encountered and processed by Instant Database, the 6 choices will be displayed in a drop down list at the right side of the IDB screen. You select one of the items. Let's say it was <<jpwsig>>. The characters <<jpwsig>> will be placed into the document in lieu of [*Signature*]. Pathagoras will then immediately hunt down the value for the term "jpwsig" and place it into the document.

By thoughtful use of variables and masks, you can transform plain Word documents into powerful interactive and automated templates, all with plain text and no hidden fields.

See also:

[<<Folder Function>>](#)²²⁸

[Double Angle Bracket Functions](#)¹⁰⁴

[Order of Search Rules](#)⁹⁵

[Multichoice *List*](#)⁴⁵⁸

[Instant Database Mask](#)²⁶⁰

[Clause Sets](#)¹⁰⁴

7.13 <P>aragraph Function

If you wish your replacement text to span more than one line (perhaps to put an address block with Name, Address, City, ST ZIP), type '<P>' or '<p>' where you wish a line break to occur.

- Examples:

Variable (in document and
on left side of IDB Screen)

Replacement Text
(right side of IDB Screen)

[Name & Address] John Q. Public<P>1234 Main Street<P>Boston, MA 23454

[Signature Line] Sincerely Yours,<p><p><p>Barack H. Obama<P>President, United States

[Property Desc] <<C:\data\property descriptions\Bush, G\description.doc>>

[Name & Address]

Thank you for your interest in [Property Desc].

It is not currently for sale at any price, but under certain circumstances you may be allowed to live in it for free. Call me for details. Let me know if you have any other questions.

[Signature Line]

will become

**John Q. Public
1234 Main Street
Boston, MA 23454**

Thank you for your interest in 1400 Pennsylvania Avenue, Washington, D.C. It is not currently for sale at any price, but under certain circumstances you will be allowed to live in it for free. Call me for details. Let me know if you have any other questions.

Sincerely Yours,

**Barack H. Obama
President, United States**

7.14 <MailTo:> function

If you wish your replacement text to be, or include, a link to call up you (or your recipient's) mail handler, pre addressed to a recipient, use the <MailTo:. . .> function. The setup is simply the open angle bracket followed by the command MailTo: (colon is mandataory), the email address (in standard a@b.com style), followed by a closing angle bracket.

Examples:

In body of source document:

[<Mailto:info@pathagoras.com>](mailto:info@pathagoras.com)

In a 'normal' document: Type as shown above. (The blue coloring is not required. It is for emphasis only.) The conversion will take place after the Instant Database is run (just like, and along with, the 'spell-out' and other functions). It is unlikely that you will use this function in a 'normal' document because the email address can be embedded in the document from the outset. The more likely application is, as shown below, within a MultiChoice *List*. That is because the

Within a MultiChoice *List*:

Let's say the alias is "*Attorney Name*" and you want each choice to include the attorney's name, address and email address, all 'stacked'.

You might have the following entries:

Adam A. Able, Esq.<p>123 Main Street<p>Hampton, VA
23669<p><mailto:aaattorney@ablebaker.com>/Betty B. Baker, Esq.<p>123 Main
Street<p>Hampton, VA 23669<p><mailto:bbbaker@ablebaker.com>/Carl C. Charles,
Esq.<p>123 Main Street<p>Hampton, VA 23669<p><mailto:cccharles@ablebaker.com>

Reminders: <p> is plain text for 'insert paragraph'.

The blue coloring in the example is for emphasis. No coloring is required.

If the first choice of the MultiChoice *List* is selected, the result will be:

Adam A. Able
123 Main Street
Hampton, VA 23669
aaattorney@ablebaker.com

7.15 <URL:> function

If you wish your replacement text to be, or include, a link to a website or to download a file, use the <URL: . . .> function. Examples

<URL:www.pathagoras.com>

<URL:www.pathagoras.com/MiniManuals/Creating Variables.pdf>

Examples:**In body of source document:**

[<URL:info@pathagoras.com>](http://info@pathagoras.com)

In a 'normal' document: Type as shown above. (The blue coloring is not required. It is for emphasis only.) The conversion will take place after the Instant Database is run (just like,

and along with, the 'spell-out' and other functions). It is unlikely that you will use this function in a 'normal' document because the email address can be embedded in the document from the outset. The more likely application is, as shown below, within a MultiChoice *List*. That is because the

Within a MultiChoice *List*:

Let's say the alias is "*Attorney Name*" and you want each choice to include the attorney's name, address, an email address, and the firms URL, all 'stacked'.

You might have the following entries:

Adam A. Able, Esq.<p>123 Main Street<p>Hampton, VA
23669<p><mailto:aaattorney@ablebaker.com><p><URL:www.ablebaker.com>/Betty B.
Baker, Esq.<p>123 Main Street<p>Hampton, VA
23669<p><mailto:bbbaker@ablebaker.com><p><URL:www.ablebaker.com>/Carl C. Charles,
Esq.<p>123 Main Street<p>Hampton, VA
23669<p><mailto:cccharles@ablebaker.com><p><URL:www.ablebaker.com>

Reminders: <p> is plain text for 'insert paragraph'.

The blue coloring in the example is for emphasis. No coloring is required.)

If the first choice of the MultiChoice *List* is selected, the result will be:

Adam A. Able
123 Main Street
Hampton, VA 23669
aaattorney@ablebaker.com
www.ablebaker.com

7.16 <x.> Function Reminders

You can get a 'reminder' of the <P>, <S> and <\$> functions discussed above directly from the Instant Database screen. Just click the "Show <#> Tips" button just below the last replacement text box. A box will pop up with examples and usage tips.



The Pathagoras System

Instant Database Records

Part



VIII

8 Instant Database Records

After you have completed the Instant Database form, you are presented the question "Do you want to save the contents of this Instant Database screen as a new Data record?"

If you answer this question "Yes", you will be asked to provide a name for the new record. Once you provide the name and click 'OK', the data on the screen is saved into a file bearing the name you just provided. Once saved, the data can be recalled and reused any number of times. This is the 'database' aspect of the Instant Database module.

A couple of pointers:

- The name you provide can be anything. Your client's or customer's name; a file number; a combination of the two.
- The entries in the Existing Data dropdown list are presented in alphabetical order. Therefore, if you use a client or customer's name as the record identifier, we recommend the "Lastname, FirstName" style.
- Many law firms represent major clients, and they may wish to store records in sub-folders by client name. Pathagoras allows this in its 'tree service.' You can create a small 'tree' in the same folder in which data records are originally stored and locate records a bit faster because of this added level of organization.
 - A 'tree' is simply a sub-folder beneath the 'main' data folder.
 - To activate 'tree service' for your records, click Utilities/Settings from the Pathagoras drop down features list. Click All Settings and then select the Instant Database tab. Click the box labeled "Enable Tree Service for Data Records."
 - Once tree service has been enabled, the next time you save a Data Record, you can select (or create) the sub-folder into which you wish to save the record.

8.1 IDB Files Location

The current physical location of your Instant Database Files is displayed in numerous locations. The easiest to see is at the bottom of the Instant Database screen after you press <Alt-D>. You can also see it by clicking the entry marked "File Locations" of the 'All Settings' screen (look for 'Instant Database').

You can easily re-point the current pointer to a different location. Do so, of course, if you have moved (or desire to move) your records to a new location.



If you have a single Pathagoras license, you will not likely be changing the location of your Instant Database records. But if you have purchased the Network version of the program, or otherwise want to make your records more accessible (perhaps cloud based), you may wish to share records via a common folder. To share records, you only need to re-point Pathagoras to a common location. (And if the records aren't yet in that common location, don't worry. Pathagoras will help you to move them.)

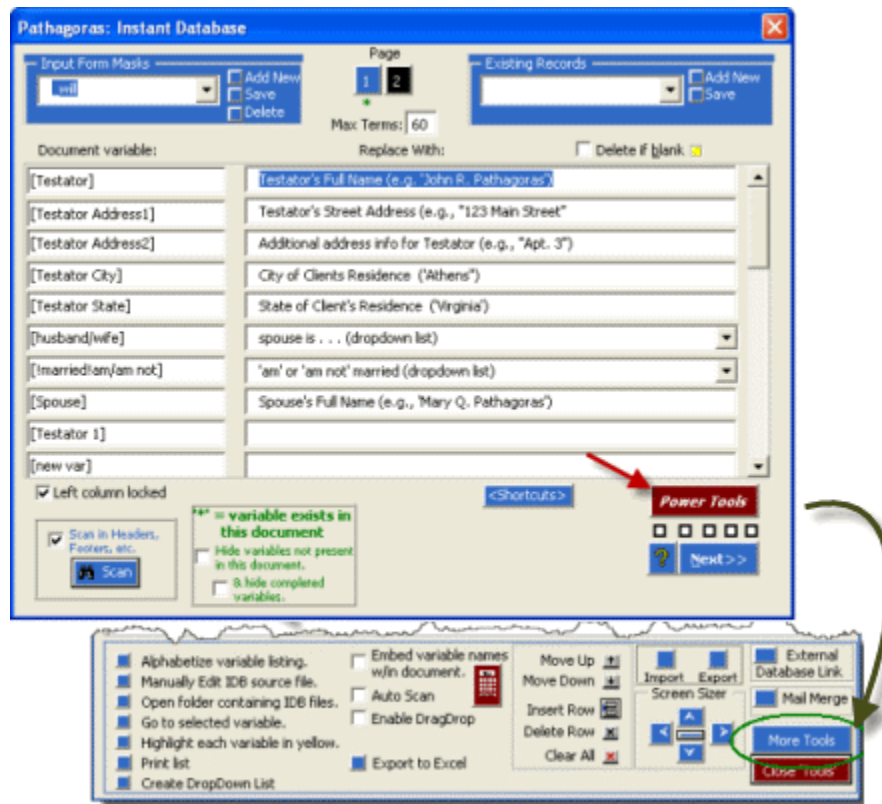
This 're-pointing' screen discussed above can be accessed in three equally acceptable ways.

- a) **Via the face of Instant Database screen:** Shift-click on the text box that shows the current location of the files.

OR

- b) **From the IDB/Power Tools:**

1. Display the Instant Database screen (Alt-D) and click the red <Power Tools> button.
2. Click the <More Tools> button on the resulting screen.



OR

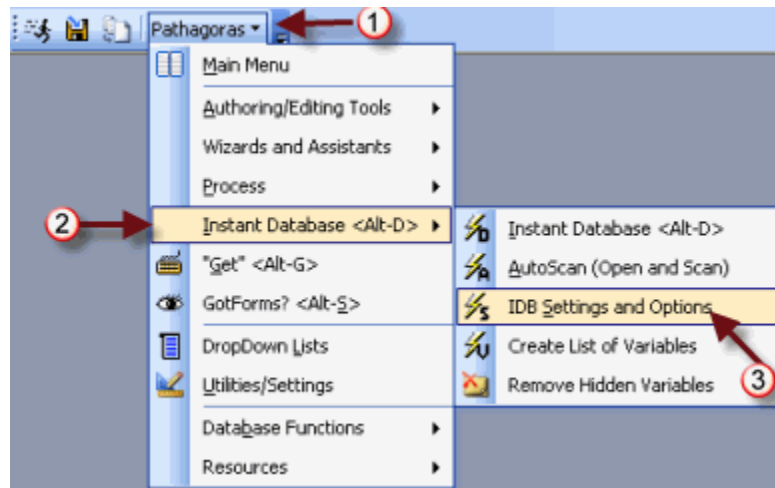
- c) **From the Utilities and Setting Screen**

1. Click the Utilities/Setting entry from the Pathagoras drop down features menu.
2. Click the All Settings' button on the front page of the resulting screen.
3. Click the Instant Database tab.

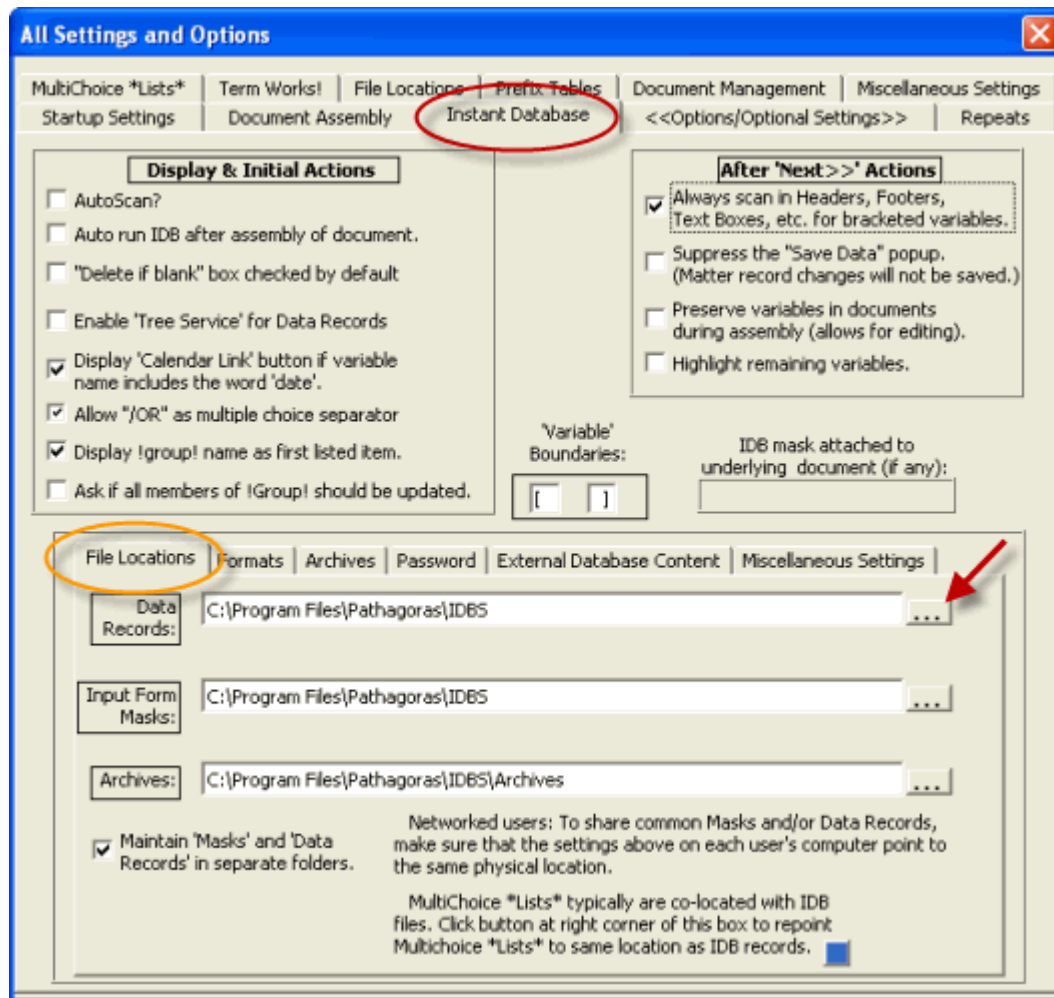
OR

- d) **Via the Pathagoras drop down features list:**

Click the *Instant Database* entry and then click *Instant Database Settings and Options* from the sub-menu.



All 'roads' (above) lead to this screen:



IDB Tools (File Locations tab exposed).

It does not matter where you locate your Data records or Input Masks,
or what you call the folder in which they are saved
(although we do recommend using 'IDBs' as part of the path name).

If you wish to use a common set of records among all users, all users must point to the same location.

The default location of the Instant Database records is (as shown above) "C:\Program Files\Pathagoras\IDBs". If you have a single Pathagoras license, you will not likely be changing the location of your Instant Database records. But if you have purchased the Network version of the program, or otherwise want to make your records more accessible (perhaps cloud based), you may wish to share records via a common folder. Here are the steps:

1. Display the IDB File Locations tab following any of the steps outlined above.
2. In the field called Data Records, manually type the full path to the 'common' folder into the text box next to "Data Records"; **OR**
3. Click the navigation button [. . .] at the right side of the screen adjacent to the Data Records text box. Follow the prompt and navigate to the desired folder.
4. When finished, click the Save icon to lock in your changes.

NOTE: The Input Masks and Data Records can be stored together or stored separately. It is a personal choice. (We recommend together, and that is the default setting.)


NOTE 2: There simply are no rules or 'best practices' when it comes to sharing common Instant Database records. It all depends upon how your office is run. Some offices share everything, some offices share masks but not client or customer data records. Some offices share nothing, and run all instances of Pathagoras as totally independent operations.

8.2 Sharing Data with other programs

Pathagoras saves Data Records as individual "CSV" files. CSV stands for "comma separate values," and is a common format for sharing information among unrelated programs. (More accurately, Pathagoras uses a 'quote-comma' scheme to separate fields. Individual fields are enclosed in double-quotes and separated by commas. This information is typically automatically detected by most programs.)

Pathagoras can read CSV files that may be created by other programs following the quote-comma field separation scheme, and other programs can read the CSV files that Pathagoras creates.

The folder into which Pathagoras saves IDB records can be found in the File Locations tab of the Instant Database Tools screen. Access is via the Pathagoras dropdown menu in the tool bar. Point your external database to this folder to accomplish the data share.

 On all versions prior to 12.1 build 20, you have to affirmatively select the 'Save Using CSV' option. Here are the steps:

1. From the Pathagoras Features Menu, select Instant Database|Instant Database Tools
2. On the 'front' tab, scroll down to the "Data Saving Format" section. Select CSV. ('Both' is optional, but not recommended.)

Pathagoras has the ability to link to external data source (Access, Excel, Act, Alpha5 and a myriad of other databases. It is not a supported feature of Pathagoras, but detailed information on how to implement such 'external database' linking can be found [here](#)²⁸⁶, and at the links pointed to on that page.

8.3 Archiving IDB Records

As you accumulate more and more IDB records, you may wish to save out the older ones (or delete them if they will never be needed again). The tools which allow you to do so can be found under the Archives tab of the Instant Database Tools screen. Access IDB tools from the Instant Database entry of the Pathagoras features menu.

8.4 Easy As Pie IDB Records

Enter topic text here.

8.5 Print IDB Data from Multiple Records

Pathagoras lets you create a listing of up to 6 data elements from your Instant Database Records.

Uses:

- Create a list of Names and Addresses (assuming that addresses are maintained in your IDB records).
- Create a list of names and scores (or prices, or quantities) (assuming that scores, prices or quantities are maintained in your IDB records).
- Create a list of any value that you maintain in your IDB records.
- Use to create reports and even analyze data. The tables you create here can readily be transferred into Excel for analysis in many views..

How to:

Two methods exist by which you can create a table

1. From the Instant Database Setting & Options Screen.

- Click the "Convert/Create" tab from the bottom set of tabs.
- (Optional) If you have activated the IDB 'Tree Service', select (either by typing or by navigating) the folder from which the table will be created.

All Settings and Options

MultiChoice *Lists* | Term Works | File Locations | **Prefix Tables** | Document Management | Excel | Miscellaneous Settings

Startup Settings | Document Assembly | **INSTANT DATABASE** | <<Options/Optional Settings>> | <<Repeats>>

Display & Initial Actions

☐ AutoScan after Alt-D?

☐ Auto run IDB after assembly of document.

☐ "Delete if blank" box: checked by default

☐ Process Options/Optional and Repeats blocks before displaying IDB screen. ☒ But ask first

☒ Allow "/OR/" as multiple choice separator

☐ Display IgroupI name as first listed item.

☒ Auto Re-scan after call of existing data record.

After 'Next>>' Actions

☒ Always scan in Headers, Footers, Text Boxes, etc. for bracketed variables.

☐ Suppress "Save Record" popup. (Data records will not be saved.)

☐ Preserve variables in documents during assembly (allows for editing).

☒ Highlight remaining variables.

☐ Process remaining <<*Options/Optional & Repeats*>> text blocks

Mask Maker

☒ Create a Mask (Input Form) automatically from a folder of documents.

Archives | Custom Settings | External Database Settings | Miscellaneous Settings | **Convert/Create** | Password

☒ Convert 2 column table (or two columns from table) into Instant Database Record.

Create/Replace

Folder containing records: c:\Program Files\Pathagoras\IDB5\

Create table of IDB records containing these variables: (Row 1 first, then second row)

[First Name] [Middle Name] [Last Name]

Rename Old Variable with New Variable in Existing Records

Replace: [Name of Client] with [Client Name] ☒ Replacements will be made in all records in folder designated above.

2. In-line:

- a) Type the formula using this style in your document at the point you want the table inserted.

<<*IDBTable*variable1/variable2/variable3/variable4/variable5/variable6>>

(If you want just the table, type the above on a blank document.)

- b) **Example:**


<<*IDBTable*[First Name]/[Last Name]/[Street Address]/[Sales Price]>>


- c) **Anatomy:**


Call to action: <<*IDBTable*

Variables list: List up to 6 variables, each separated by slashes.
The variables must include the surrounding brackets, if used.

Closing mark: >>

 The colors used in the above example are for illustration and emphasis only. Colors are NOT required by the formula.

 If you have activated the IDB Matter 'Tree Service' (allowing you to store IDB records in sub-folders), you will be prompted to navigate to the folder in which the records you desire to list reside. Otherwise, Pathagoras will print data from each record in the main IDB folder. (Currently you cannot print records from multiple folders, or from sub-folders.)

 You can sort the resulting table using Word's table tools. If you need to do more elaborate financial analysis on the results, consider doing that in Excel. To move the data into Excel, simply highlight the table (or as much of it as you want to analyze) and copy. Open Excel and paste. It is as simple as that!

8.6 Changing Variable Names in Existing Records

Possible scenario: When you first started working with Pathagoras, you called your variables “[Name of Client]” and “[Address of Client]”. (Note that when these sort alphabetically, they likely will not appear next to each other.)

Now that you have more experience with Pathagoras, and your system has matured, you have decided that you want your variables names such that the information 'groups together' better. You want the variables to be “[Client Name]” and “[Client Address]”.

You can make the change in future source documents by using “[Client Name]” and “[Client Address]” from the outset. You can change existing source documents to reflect “[Client Name]” and “[Client Address]” using Pathagoras' [Search and Replace](#) ⁴⁴⁸ tool. (Note: Search and Replace will work against all documents in a selected folder, so the replacement can be done quite rapidly.)

Having done that, you now need a tool that allows you to make your existing data records compatible with your new documents.

You can replace a (now) 'bad' variable name with a 'good' variable name using another one of

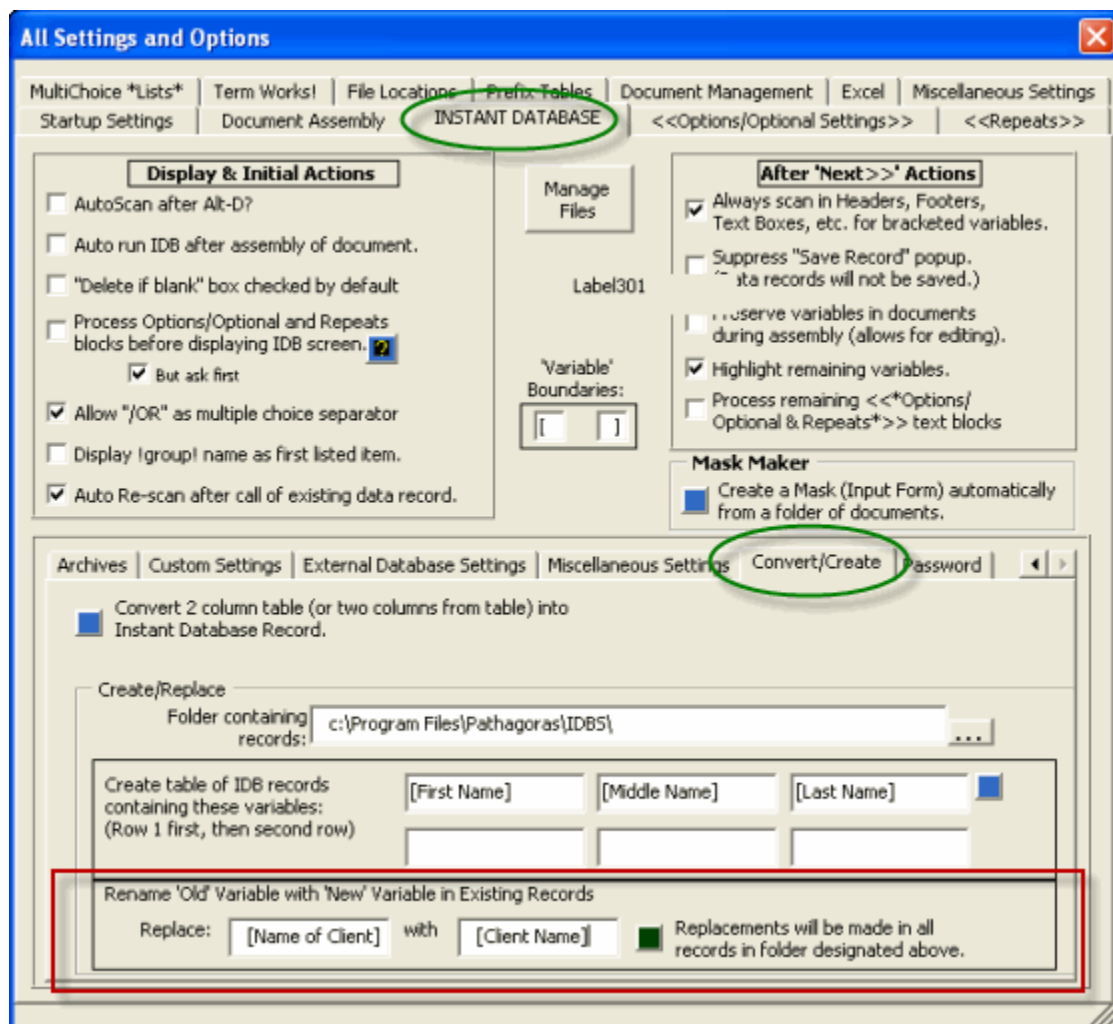
Pathagoras' replacement tools. This one will inspect your entire collection of records and (following the above example) change "[Name of Client]" to "[Client Name]" in each record. You can change up to 3 variables at a time.

Here's how:

From the Instant Database Setting & Options Screen.

- Click the "Convert/Create" tab from the bottom set of tabs.
- (Optional) If you have activated the IDB 'Tree Service', select (either by typing or by navigating) the folder from which the table will be created.
- In the lower set of fields, type the old variable(s) in the top row and the new variables in the bottom row.

(The below image is a bit outdated. You can now replace at one time up to 3 'old' variables with new ones.)



The Pathagoras System

Intake Forms

Part




IX

9 Intake Forms

Pathagoras offers several ways for you to gather data for use in the Instant Database system.

- The most popular method is the one previously discussed. First assemble a document. Then press Alt-D to activate the Instant Database system. With the Instant Database screen visible, press the 'Scan' button. Pathagoras will quickly locate all bracketed variables and display them in left column of the screen. Provide replacement values for each of the variables. When you click the Next button, Pathagoras will not only replace the variables with the values you provided, but will save the variables-to-values pairings in a database for reuse.
- Sometimes you want to create a database record for a client or customer without assembling a document. In other words, you want an Intake Form in its purest sense. Here are two approaches using the Instant Database screen as a starting point.
 - Create a document that contains all of your variables and nothing else. Recall the document, press Alt-D to display the Instant Database screen. 'Scan' the document. Pathagoras will display the variables in the left side of the IDB screen. Complete the right side and press the Save button. (Press the Next button if you want Pathagoras to replace the variables in the document with the values. That way you will have a printout of the values. Be sure not to overwrite the original document.
 - Create an 'Input Form Mask'. This is special Instant Database record that has been specially designed to gather data most efficiently. Masks can contain completion tips, multiple choice variables, and all other manner of assistance in inputting the data you need. A whole section of this Manual is dedicated to the design and use of Input Form Mask.
- Pathagoras provides several other methods for collecting client and customer data that are 'alternatives' to entering data directly onto the Instant Database screen. These techniques are designed such that you can ask your client or customer to complete the intake form on their own computers. This will usually save you time (because someone else is doing the heavy typing), eliminate spelling and other errors, and get the client/customer invested in the process.

Once the completed form has been returned, Pathagoras provides the tools to 'scrape' the data from the form. The result is an instantly usable Instant Database record with less effort on your part.

Two of these methods use Word-based forms. You will find them easy to create. They are discussed on the pages that immediately follow. A third method (using Excel as the data collection tool) is discussed at [this link](#)  616.

9.1 Instant Database 'Input Form' Masks

Instant Database: 'Input Form' Masks

What is an 'Input Form' Mask?

An 'Input Form' Mask is simply a data entry form.

- A data entry form is an device by which an operator tells the program how to replace document variables with personal information.
- Every document assembly program requires a data entry form of one kind or

another.

- The input form discussed in this Manual up until this point is the form created when you press Alt-D to display the Instant Database screen. When you press Scan, Pathagoras populates the IDB screen with the variables in the underlying document. But if you want a pre-designed form, with the variables occurring in a pre-set order, with (optional) hints and tips and validations, you can design your own. And that would be a 'mask.'

What distinguishes Pathagoras' data entry form from those of its competitors is the ease by which Pathagoras can generate them and by which you can use them.

- With almost every other program, the data entry form must be pre-constructed by the system administrator and 'locked down' before it can be used. Without the intake form, the document simply cannot be personalized. In some cases, the document cannot even be created without first completing the intake form. Not so with Pathagoras.
- Pathagoras data entry is fluid and flexible. All is done from within Word without remote tables, meta-data, fields and code. The mask we propose here is based on what likely is an already familiar Instant Database screen.
- A Mask is an optional approach. If you have created no Mask, the system (including personalizing the assembled document) will still work without a hitch. (You know that because you have probably already personalized dozens of documents without a mask, using the basic features of Instant Database.)

Why create an 'Input Form' Mask?

- **Primarily, you need a mask to tell Pathagoras what personal information is to replace the document variables.**

Without a tool for the end user to tell the program what values should be substituted for each document variable, it would be quite difficult to 'personalize' the document for the ultimate client or customer. (The computer operator could perform a manual 'search and replace' operation for each variable, but this is not a satisfactory solution.)

This 'need' explanation is only partly satisfactory because Pathagoras is unique among all programs in that a data collection form can be easily created simply by pressing Alt-D and the clicking the Scan button. So you don't 'need' a mask to tell Pathagoras what personal information is to replace the document variables. But you should consider introducing them into your maturing system nevertheless. Read on.

- **Masks allow you better data input control.**

With a 'standard' scan, Pathagoras displays the variables in the order they appear in the document. Often, that order is not the most logical order for inputting personal data into a worksheet.

With a mask, you can organize the entries in a definite order, perhaps one that mimics your client or customer intake sheets, standard purchase orders, etc. That way you can insure that data entry will occur in an organized, logical fashion for users who aren't familiar with Pathagoras.

See [Organizing Mask Entries](#) 

- **Masks allow you to provide completion tips and examples to the operator.**

Masks can also help make the IDB process more user-friendly for the end-user by providing completion tips and examples.



See [Making the Mask More Meaningful](#) 

- **Masks allow data input independent of document assembly.**

Masks are useful when the office protocol is to complete the personal data before building the document. (Without a mask, you must have an underlying document to scan for variables. Maybe you want someone to record values independently of scanning or completing a form. That is precisely what Masks allow. Instead of scanning a document for variables, just click on a Mask you have created. Complete the values, save and move on.

- **Masks make it easy to Pathagorize existing or new documents.**

Call up a mask while you are 'Pathagorizing.' Then you can either



- ['drag and drop'](#)  a variable directly from the mask into the document you are editing, or
- create a [drop down list](#)  of the mask's variables and point and click any variable into the document you are editing.

- **If you call data in from external data sources (Excel, TimeMatters, etc), Masks allow you to control the order and scope of the values called in.**

- Most external databases contain values that are far in excess of the variables needed for daily document assembly.
- Masks allow you to call in and work with just the values your wish.
- Using the concatenate functions you can build document variables (e.g., [Client Name] from bits and pieces of the values found in the external database.



You likely will need several masks for a complete document assembly system. The variables that are needed to complete, let's say, a 'Will' are significantly different from the variables needed to draft a 'Contract' or a 'Purchase Order.' It is perfectly fine (indeed, it is a 'best practice') to have a variety of masks, one for each genre of document or office practice (estate planning, contracts, litigation, bids and RFPs, etc).

The Mask is but on technique for gathering client and customer data. Check out [Word-Based Intake Forms](#)  and [Excel Intake Forms](#)  for other techniques.

9.1.1 Creating a Mask

You can create an Instant Database Input Form Mask either manually or automatically:

- **Manually:** Display the Instant Database screen by pressing <Alt-D>. Manually type the known, anticipated or desired document variables into the left column. Create mask by using the 'Save' button in the upper *left* side of screen. (Image below.)
- **Automatically, via Mask Maker:** Mask Maker will scan one, several or an entire folder's worth of documents for variables. It will put those variables onto a screen where you can edit, rearrange and save out as a Mask. Read more about Mask Maker

on the [next section](#)²⁶⁵ of this Manual.

- **Automatically, alt.2:** Open or create a document that contains many variables. Display the Instant Database screen <Alt-D> and then hit the blue Scan button in the lower left hand corner of the IDB screen. Pathagoras will locate all bracketed variables and display them as Document Variables in the left column. Save.



Tip: To make sure that Pathagoras gathers as many variables as possible during the scan described above, consider assembling one huge document containing at least one of every representative document in the genre or class of document for which you wish to create this mask. That's right! Assemble the largest Will and the most complex Power of Attorney and the biggest Trust all as one humongous document. (Just assemble it. You won't need to save it.) Then Scan the document. Rearrange the variables in a "makes sense" order using the Up and Down arrows. If variables are not present, add them.

- **Another alternative:** Call up an existing client or customer's record from the Existing Data records list. This should be a record that is 'mature' (i.e., it currently contains all or at least the major portion of the variables you typically use in documents of this genre. Change the personal (replacement) text showing at the right with instructions or qualifications you want to provide to the end user. I.e., Modify 'John Q. Doe' on the [Customer Name] row to read Customer's Full Name [e.g., 'John Q. Doe']. Save as mask using the 'Save' button in the upper *left* side of screen. (Image below.).

➔ **Save often:** Save the Input Form Mask by clicking the "Add New" (the first time you save) or "Save" buttons in the "Input Form Masks" section in the upper left hand corner of the Instant Database screen. (Figure 1, below.) If you are saving the mask for the first time, provide an appropriate name that reflects the 'genre' or subject matter of the documents in which you will be using these variables. E.g., "Estate Planning" or "Proposals" or "Litigation," etc. See discussion on 'genre' below.

Click the Masks|Save button to save your new (or updated) mask.

Notes and Final Thoughts:

Number of Variables. Each Input Form mask can include an unlimited number of Document Variables. The default number of Document Variable is set at 60; however, you can change it to any other number by simply changing the number in the Max Terms box at the right side of the screen.

Overbuild with Variables. Input Form masks can be (and actually should be) “over-built” to contain every variable that you potentially might use for the document’s genre. In this context, ‘genre’ means ‘general subject matter.’ As said before, the variables in a Will will be quite different from the variables in a Purchase Order, but the variables in a Will may overlap those in a Trust and Power of Attorney.


Maximize Consistency within Genre. The ‘humongous document’ approach described above also lets you see if your variable names are consistent from document to document. For example: If the same variable is [ClientName] in a Will and [Client Name] in a Power of Attorney, pick one approach and use it consistently in both documents. They are of the same genre, and should share a common Mask and use identical variables.

Minimize Variables within Genre. Your goal should be to reduce the number of individual variables to the smallest number possible for each document genre. Spend some time up front to rename similar variables with a more generic term that can be applicable to all of them. For example: A “Testator” of a Will, a “Settlor” of a trust and the “Grantor” of a Power of Attorney are frequently the same person in an estate planning package. Since the documents are of the same genre, rename them something like “[Adult1]” instead of a topic specific name, so that the

variable is completed only once. On the mask, you can give the end-user very specific guidance and instructions as to who or what 'Adult1' stands for.

OR

Use Equivalency Function. As an alternative to the immediately preceding paragraph, consider using the equivalency function to avoid having to type the value of related variables more than once. This function allows you to maintain multiple variables that refer to the same actor. The value of the 'extra' variable is made equivalent to the value of another variable.

Click this link to read more about the [Equivalency Function](#)  230.

Number of Masks. You likely will need several Input Form masks for a complete document assembly system. This is because the variables typically found in, let's say, a Will are probably quite different from those found in a Purchase Order. It is perfectly fine (indeed, it is a 'best practice') to have a variety of masks for a variety of situations. That said, a mask should be written in such a way that it can support multiple documents and source forms. Consideration should be given to using a common variable naming scheme across a range of similar or related documents. For example, wills, trusts and powers of attorney might contain similar naming schemes.

Modifying Masks. It is easy (very, very easy) to modify a mask. Just recall the mask and type the changes on its face. Rearrange the fields if appropriate. Don't worry if you don't have every variable in your mask when you save it. Don't worry if the order of the clauses in the mask is not the 'best.' Pathagoras lets you easily recall it and easily modify it later.

- An existing Input Form mask can be temporarily or permanently augmented if the current document has variables not contained in the mask. Simply press the Scan button. Any new bracketed variables will be added at the end of the list. If you wish, you can edit and save the augmented listing.

See Also:

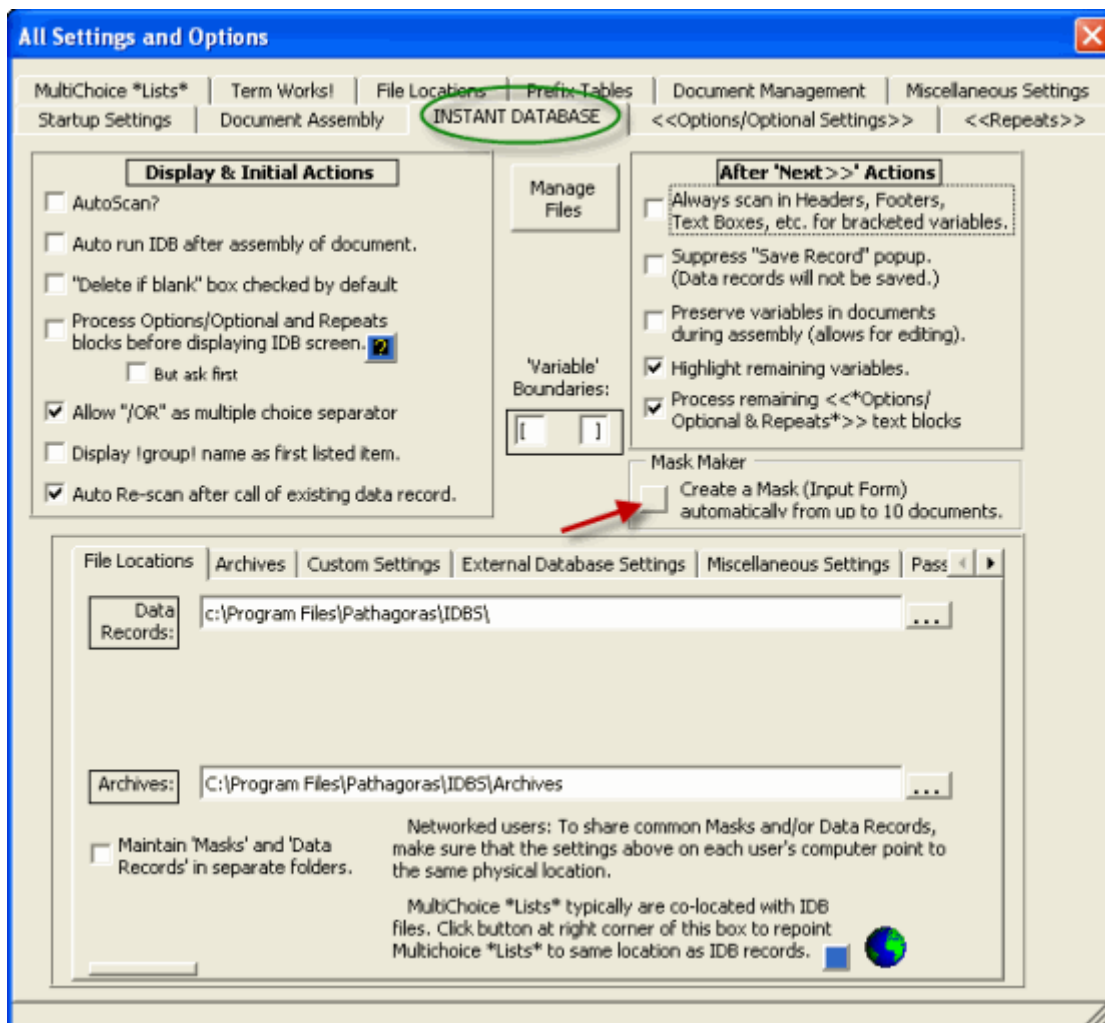
[Making the mask more meaningful to the end user.](#)  267

9.1.2 Mask Maker

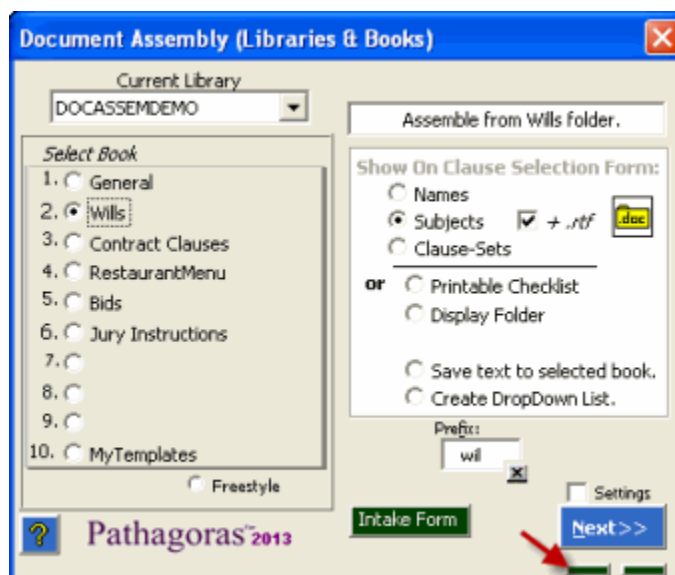
You can create an Instant Database Input Form Mask automatically using the Mask Maker tool.

You can activate Mask Maker from two locations:

- On the Instant Database Settings and Options Screen (right side, MaskMaker frame)



- If the folder you want to read is a book, click the book in the Document Assembly screen. Its the leftmost green button underneath the Next>> at the bottom of the screen.



- MaskMaker is also a QuickPicks/My Buttons option.

To use: Open a document that contains variables. Click the Mask Maker button from the All Settings screen (Instant Database tab.) Scan underlying document and up to 9 more from same folder for variables. Choose whether you want to scan just the underlying document for variables or all (up to 10) documents in the folder for variables.

Note: You cannot edit from the Mask Maker screen. It is designed to gather variables from the designated documents. But it will show you (and allow you to open) which document in which the variable made its first appearance. That way if you see a variable that you need to change and click the GoTo button. Pathagoras will display the document and allow you to make corrections. (So Mask Maker will help you to assure consistency of your variables across a folder-full of documents.)

Edit any Masks you create by recalling them using regular Instant Database screens and tools <Alt-D>.

9.1.3 Making the Mask More Meaningful

Making the Input Form Mask more Meaningful.

When you created the mask (whether you manually typed in the variables or called them in automatically with a Scan), you ended up with variables in the left column and nothing in the right column. You can use that blank space to make the Input Form more 'user friendly' by inserting a '*completion tip*' which can provide useful information to the end-user.

Example: The variable [Date] might appear in your document and therefore in your mask. But what does [Date] mean to the end-user who might be seeing this variable for the first time?

- Even if the user could figure out to what 'Date' refers from its context, what style of date is expected? Full date, abbreviated date, numbers only?
- You can easily make the screen 'friendlier' by creating a mask and typing in to the right of the [Date] (or any other) variable a 'completion tip.' Continuing with the example, the following would work: "Date of Offer (e.g., "March 24, 2009")". This tip will be saved along with the variables and displayed when the mask is called up again.
- During an actual document session, the tip will advise the operator not only to what 'date' the word "Date" is referring but it shows the 'style' of the date that is being requested ("March 24, 2009" as opposed to "03/24/09" or any of the half-dozen other possible date formats.)

See below for illustration.

The otherwise 'blank' space at the right need not be wasted.
The above is an IDB mask with 'completion tips' provided at the right.

➔ Don't forget to save your work!

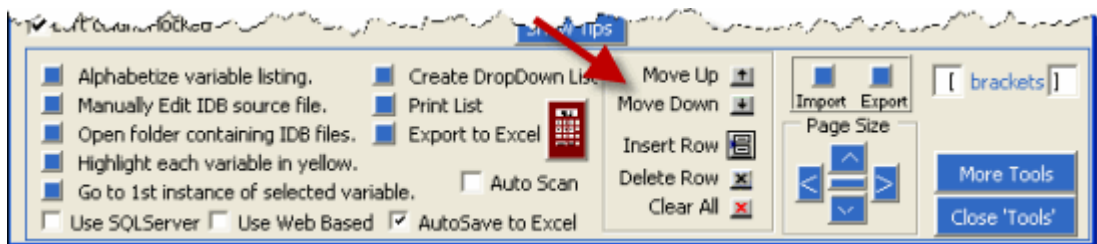
9.1.4 Organizing Mask Entries

Organizing and Changing a Mask

Once you have created the basic mask, either manually or automatically, you may want to reorganize or make other adjustments to make it more consistent with your intake sheets or more logical for your staff to complete.

- **Rearranging, Inserting, Deleting:** Using the buttons found in the **Power Tools** section of the screen, you can easily arrange and rearrange the order of the variables, insert additional variables, and delete variables you don't want.

When you are satisfied with your changes, save your mask again by hitting the "Save" button in the "Input Form Masks" box in the upper left hand corner of the IDB screen.



Using the Move Up/Down Arrows, the Mask can be made more 'user-friendly.'



Don't worry if every variable is not in your mask when you save it, or if the order of the Document Variables in the mask is not the 'best.' Pathagoras lets you easily recall it and easily modify it later. Simply open the IDB screen, recall the mask from the list at the left, make your desired changes, and save it again.



Sorting the variables: You can manually 'sort' your variables by moving them up and down on the Instant Database page. A faster way, when you have a lot of variables to sort, is to preface the variable with a number and let Pathagoras do the sorting for you. After you have numbered the variables, click the 'Alphabetize Variables' item in the Power Tools page.

Note the following: The sort is alphabetical. To sort numbers 'alphabetically' requires that each number be the same length. Otherwise '2' will sort after '10'. So, if necessary, pad the sorting # with '0's'. '01', '02', . . . '10' . . . '60', etc.; If you expect over 100 variables: '001', '002', . . . , '010' . . . , '100', etc. Once fully sorted, save the mask. Even though saved, you can freely edit the mask. Just save it again after you have finished editing.

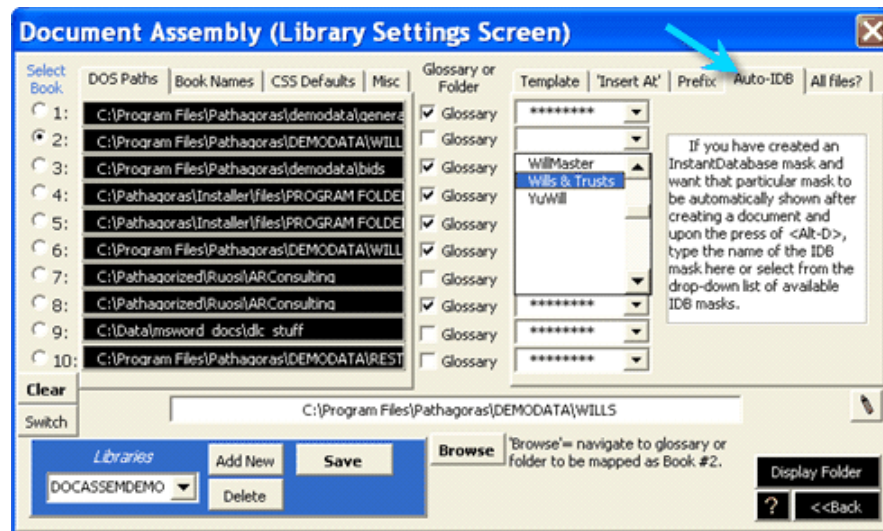
9.1.5 Assign Mask to Book

You can create a permanent link between a selected book and a selected mask. This link means that you will not have to select the mask at the end of the document assembly session. Pathagoras does it automatically.

Creating this link also allows data entry to readily be done independently of a document assembly session. See the 'lightbulb' message below.

Assign a Mask to a Book:

1. Display the Libraries & Books screen.
2. Click the <Settings> button.
3. Click the AutoIDB tab on the right side of the Settings screen. On the row parallel to the book to which you wish to assign the Mask, drop down the list of available Masks and select the appropriate one. (See figure below.)



Link an Instant Database Mask with a book.

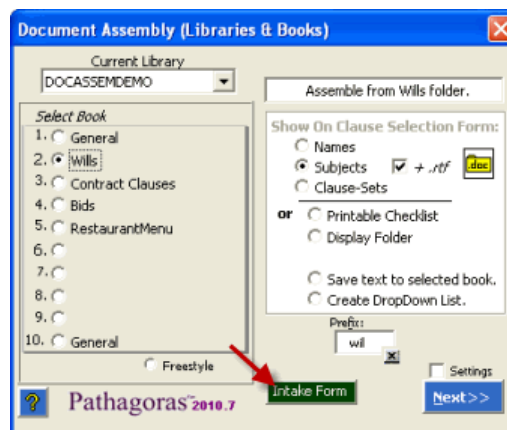
When IDB is run following a document assembly session, the Mask will automatically appear.

4. Save the <Settings> screen to lock in the association. The next time you assemble a document from that book and press <Alt-D> to personalize the document, Pathagoras will automatically display the pre-assigned mask.



There is no rule that says that personal data has to be input at a particular time. Some users wish to complete the personal data before building the document, not after. Perhaps a staff member has been assigned as the 'data-enterer' and has nothing to do with document production. Pre-assigning a mask to a book lends itself well to any scenario.

- If a mask has been linked to a book as shown above, an 'Intake Form' button will appear on the initial Libraries & Books screen. If clicked, the IDB screen appears with the proper mask displayed. Complete it, save it, and move on to something else.



The 'Intake Form' button exposed.

9.1.6 Using a Mask to 'Pathagorize' a document

Once you have created a mask, it can help you to create new documents, or to 'neuter' existing document using the variables contained in the mask.

See: [Drag and Drop: Pathagorizing using an IDB mask.](#)¹³¹
[Variable Creation Assistant](#)¹²²
[Drop down list](#)¹²⁹

9.1.7 Sharing Masks

The tools and functions applicable to distributing and sharing Matter Records apply equally to the distribution and sharing of Input Form Masks. See [Instant Database Utilities](#)²¹⁰ and [Instant Database Files Location](#)²⁵⁰.

9.1.8 Password

You can require a password in order to create or to change IDB masks. Make this selection from the IDB Settings screen.



9.1.9 Using Excel to Create Input Form

One of the most powerful aspects of Pathagoras is that it can generate a data collection form 'on-the-fly' from the variables present in the currently on-screen document. It also allows the administrator to create Input Form *masks* as described in this section when make it possible to control the order of entry of data, to provide pre-typed examples and completion tips for the end user. Masks help the data-entry person to more accurately input the client/customer/patient's personal information used to complete the assembled document. But the underlying form on which the mask is built is simple and unadorned.

Sometimes, a more extensive, universal, complex, complete, attractive, etc. intake form is needed or desired. [Pathagoras for Excel](#) can provide that. Pathagoras for Excel can generate an Instant Database record from Excel and the result is instantly usable in Word.

Here are some situations we contemplate:

- a. **Off-site collection of data.** The typical Instant Database routine in 'Pathagoras for Word' contemplates data entry on the computer on which you are working. It also

contemplates that *you* will be providing the data input labor.

Sometimes the entity from whom you want to gather data isn't within your office and/or doesn't have Pathagoras installed. In such case, the Excel solution is ideal. Just send the form over to the client or customer to complete.

Almost everyone has Excel (or a spreadsheet program that can create an Excel-type spreadsheet, e.g. Corel's Quattro Pro). You can create a customer form --let's say an intake form -- specifically designed to gather the desired data from the customer. You can send it over to the client or customer for completion (or direct to client or customer to a website where it can be completed on-line or downloaded). Pathagoras is not needed to complete the forms.

Once completed, the customer can send it back by email. You then can open the spreadsheet and use Pathagoras (Excel) to instantly move the data into an IDB record.

b. **Intake form 'design control.'** Perhaps you just want something 'prettier' or with more eye-appeal. Much fancier and user friendly intake sheets can be designed using Excel.

c. **More mathematical power and function.** Perhaps you need something more functional and powerful in terms of mathematics that only a spreadsheet can offer. Word is rather limited in its math 'skills.' Excel, on the other hand, is ideally suited for complex calculations.

d. **More control regarding validation and cascading answer sets.** In addition to powerful calculation capabilities, Excel provides very powerful validation, lookup and a myriad of other tools which promote speed and accuracy in the collection of data and selection of values. Excel allows you to select a 'top level' response (a 'category', a 'type') in one cell of the spreadsheet and by that choice alone, present to the user a limited selection of choices for the remaining cells that need to be completed. The end result of cascading answers is accuracy in result, accuracy in spelling, and less confusion by the end user as to what choices are available in a give setting.

If you are encountering a need described by any of the above situations, contact us regarding the Excel version of Pathagoras. See this page of the manual for more information, or [click here to view the separate Manual](#).

9.1.10 Importing Data Into Mask

A Mask is a primary data collection tool. That means that its primary objective is to collect from the end user the personal and specific values that will overwrite the variables in an assembled document.

Most of the time, the end user will hand type values into the Instant Database for (or click values from drop down lists displaying multiple choice options). But when set up properly, it is easy to import data from [Word documents created especially for data input purposes](#), from [Excel forms created for that purpose](#) or from [external sources](#).

Click on any of the above links to learn more, or simply continue reading this Manual. Those topics are discussed next.

9.2 Word-based Intake Forms

9.2.1 Simple Two Column Table

A simple, two-column table can be a very useful data input form. These tables are easy to create, easy to send to a client, easy to complete, and easy to process.

To create the form, simple draw a two column table in a Word document. Type into the left column the variables you want to gather from the client. (The brackets are optional. If you want to provide a form for the client or customer to complete, you may choose to omit the brackets.) Leave the right column blank. E.g.


[Client Name]	
[Client Address]	
[Client City, ST ZIP]	
[Spouse Name]	
[Child@1 Name]	

(You can copy and paste the above table into a Word document if you wish as a starter sample.)

Save the table as a Word document.

If you want the client to complete the form, email it to him or her with a request to complete the data in the right column. If an office staffer is charged with completing the form, the staffer would simply recall the form and fill in the data at the right from the client's hand written intake sheet.

The only drawback to the above form is its looks. The form looks boxy and, if you intend the client or customer to complete the form, it is not very professional looking. Further, the variable names that appear at the left may not be friendly or descriptive enough for the end user to know without assistance what is being requested. But if appearances are a secondary concern (or you plan to use the form for 'in-house' data collection only), nothing beats the simplicity of a two column table.

 **NOTE:** You can mail the form as an attachment to a email, or simply as the body of the email. (Make sure you select HTML or RichText as the email 'format.' If you send in 'plain text,' the table will be lost.)

When completed and returned by the client/customer in a reply email, just copy the table and paste it into a Word document.

Scraping the data:

When the form is complete, tell Pathagoras to create the records following these steps:

- Press Alt-D to call up the Instant Database form.
- Press the red Power Tools button
- Click the 'Import' button and select 'Import data from two column onscreen form'.
- The Instant Database screen will populate with the variable names and associated values. Continue in the normal manner to save the record.
- Don't forget: some clients are not good spellers. They don't always accurately remember

dates. You still need to check behind the informant.

9.2.2 'Fancy' Word Intake Forms

The previous pages discussed intake forms using Pathagoras "[Instant Database Masks](#)"²⁶⁰ and a 'simple, [two column table](#)'²⁷³. Both are easy to create and quite elegant in operation if they serve your purpose. However, if you wish a more attractive form that you can send to the client or customer for completion, this is where you should be.

We call the forms discussed on this page 'Fancy' because they are visually more appealing than those earlier discuss, and you can make them as fancy as you desire. The more professional you make the form, the more impressed the client/customer may be with your organization.

Definition

An intake form is any document that can be presented to a client or customer for completion. Typically, an Intake Form requests person and case related data. (Name and address of client and of individuals related to the client, dates and places related to the file, etc .) But as a practical matter any information can be requested.

When complete, the Intake Form is 'scraped' for the data it contains and a compact electronic record of the data is created.

9.2.2.1 Creating 'Fancy' Form

Creating:

There are multiple Intake Forms that Pathagoras can process. The one discussed in this section are presented as tables. Each item of data that is scraped from the form is contained in a single cell of that table. Care is taken to set up the form, but once done the form can be easily distributed, easily completed by the client or customer, easily returned after completion, and easily 'scraped.'

The size and layout of the table is your choice. As the sample form indicates, there are no size limitations. The rows need not be uniform, either in terms of size, color, number of cells, etc. You have total flexibility as to the method you present the form to the end user.

Here are the minimum requirements.

- Each cell from which you wish to gather a data item must contain a colon. The colon indicates to the end user that the cell is intended to collect information, and for 'scraping' purposes, indicates to Pathagoras where the information begins.

E.g.:

INTAKE FORM			
Your Name (First Middle Last):			
Address:			
City:	State:	Zip:	Phone:
County:	How long lived in County:		How long lived in State:

Cell Phone:	Email address:	
Date of Birth:	Birthplace:	Social Security #:

This form differs from the 'Simple' form illustrated in the previous page in several key regards:

- In the *simple* form, Column 1 must contain the name of the variables. Here, 'friendly text' and title rows can be used.
- In the *simple* form, only two columns can be used. Here, multiple columns are possible.
- Of course, the setup of a form similar to the above is more complex, but Pathagoras provides you a sample (indeed, a more elaborate version of the above) to get you started.
- While not visible in the sample above, the variable name is embedded as hidden text immediately preceding the colon in each cell which calls for data. The colon in those cells is an essential element as it separates the requested information from the 'blank' where the user will provide that information. (The colon is used by Pathagoras to determine where the data input by the end user starts.) Insert the variable names as hidden text is discussed below,

INTAKE FORM			
Your Name (First Middle Last[<i>Client Name</i>] :			
Address[<i>Client Address</i>] :			
City[<i>Client City</i>] :	State[<i>Client ST</i>] :	Zip[<i>Client ZIP</i>] :	Phone[<i>Client Phone</i>] :
County[<i>Client County</i>] :	How long lived in County:	How long lived in State:	
Cell Phone[<i>Client Cell Phone</i>] :	Email address[<i>Client Email</i>] :		
Date of Birth:	Birthplace:	Social Security #[<i>Client SSN</i>] :	

The same intake form show above but with 'hidden text' (shown by light blue and italics).
When the 'ShowAll' (¶) (pilcrow) character is clicked, the hidden text appears and disappears.

Your first form (or editing an existing form):

1. Create one or more tables in a document. As noted, the tables and their cells need not be 'uniform.' It does not matter to Pathagoras whether or not your tables look alike or whether a single row in a table has one, two or ten cells across. Mixing and matching to obtain your 'best look' is fine.

2. Put 'friendly' text in the cells requesting the desired information. Type the mandatory colon to indicate the end of the request. (The colon also denotes the start of where the end user will provide the requested data.)

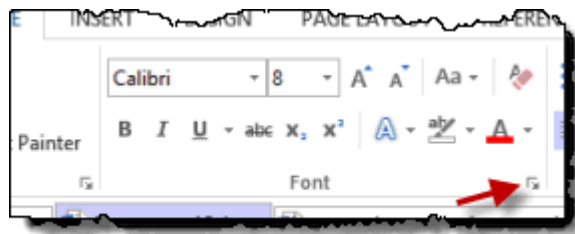
You can have title and decorative cells in the table as well. These cells should not have colons.

Once you have designed a nice looking input form, you are ready for the next step. But if you need to revise things as you move along, don't worry. You are always in Word and it's easy to change things.

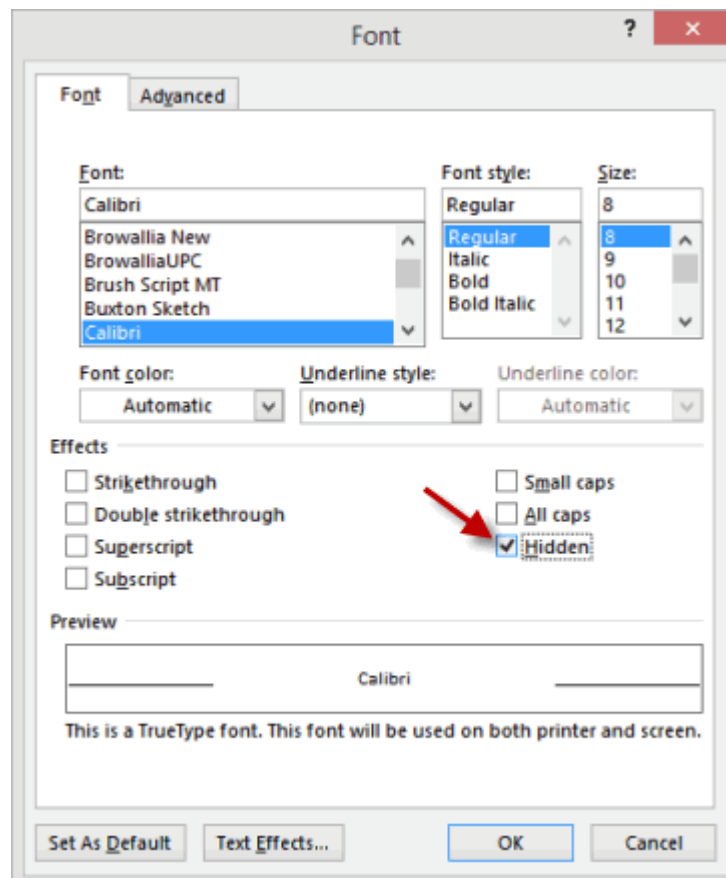
3. Add the Variable Name. Type the name of the [variable] (including the bracket) just inside (to the left of) the colon. The variable, of course, should be the variable you use in your source documents as a place holder for that information you will be gathering.

4. Convert Variable to Hidden Text: You generally will not want your end user to see the variables used in documents. So, you will want to hide them. There are two ways to convert the variables to 'hidden text.'

- **Manually:** Highlight each variable (one at a time) and convert the font to 'hidden'. Click the arrow in the lower right corner of the 'Font' section (Home tab) in the Word ribbon to turn on 'Hidden' text.

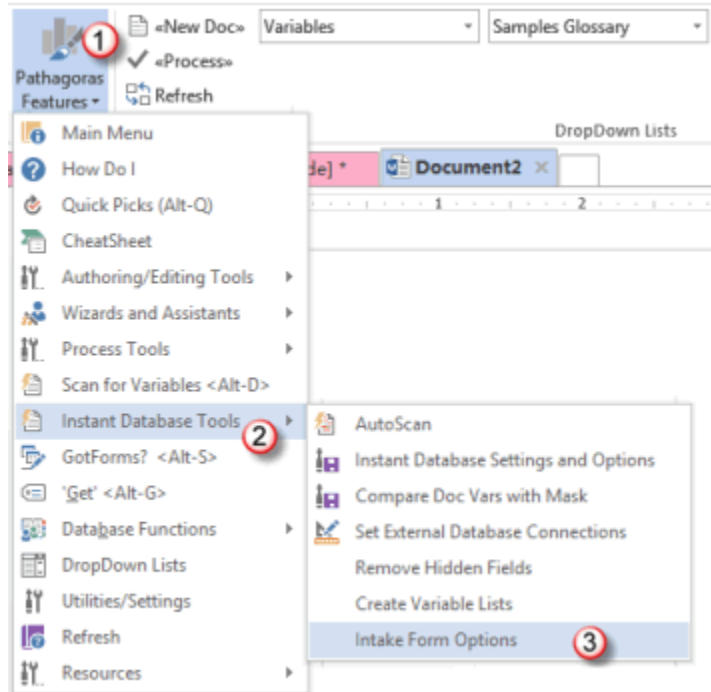


Click the small, right-downward pointing arrow to reveal the full 'Fonts' menu, shown at the right.

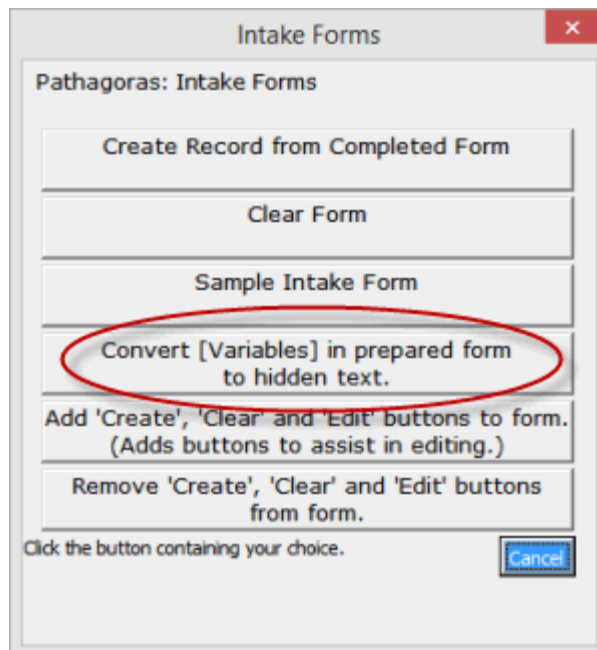


Check the 'Hidden' text box in the Effects section. (When 'visible', hidden text is indicated by small dots just below the text.) Repeat for each variable.

- **Automatically:** Pathagoras (of course) provides an automated method to convert variables to 'hidden'. Once the form is set, and the variables are placed throughout the document, click the Pathagoras features menu. Cascade down to the IDB Tools | Intake Form Options and click it to display the menu:



Click to collapse.



- **When** 'visible', hidden text is indicated by small dots just below the text.)

- To toggle the 'visibility of the hidden text, click the 'ShowAll' (¶) (pilcrow character) at the top of the editing screen (Home tab).

4. Save the form in an appropriate folder where you can easily find it. (If you send out forms frequently, you should consider making this folder one of your DropDown Lists.)

Notes:

- The name of the variable in hidden text must match the variables you are going to use for variables in your form documents. (This should be obvious, but we needed to say it anyway.)
- You should assign a hidden text variable to each value in the form that you want to capture. (Be mindful that, while every value may be important, not every value needs to be captured in your database. Your client likely won't know, for example, the case number or the judge assigned to the case, so it's best not to ask.)
- Each variable and its associated value must occupy its own cell.
- The title of the cell (the 'friendly text') doesn't matter. Pathagoras doesn't start reading until the '[' of the hidden text, parsing out the hidden variable name and the visible assigned value to its right.
- Consistent with the above, don't use brackets in your form except for identifying variables. (A workaround for this does exist, but it is beyond the scope of these instructions. If you absolutely must have brackets in your regular text, write to Pathagoras for the workaround.)
- The form can be used as a protected form. You will need to add form fields so that the tab button will jump from field to field and end up in each cell. But we recommend using it just 'as is.' The tab button will still move the cursor from cell to cell. Plus you can mouse into any cell you wish.
- Consider placing your forms in the folder reflected by a DropDown List to quickly call up a copy.

Pathagoras provides a form resembling (but more extensive than) the one shown above. You can find it in the Samples Glossary. (It contains actual hidden text, as opposed to the simulated hidden text shown in the sample.)

9.2.2.2 Scraping the Data

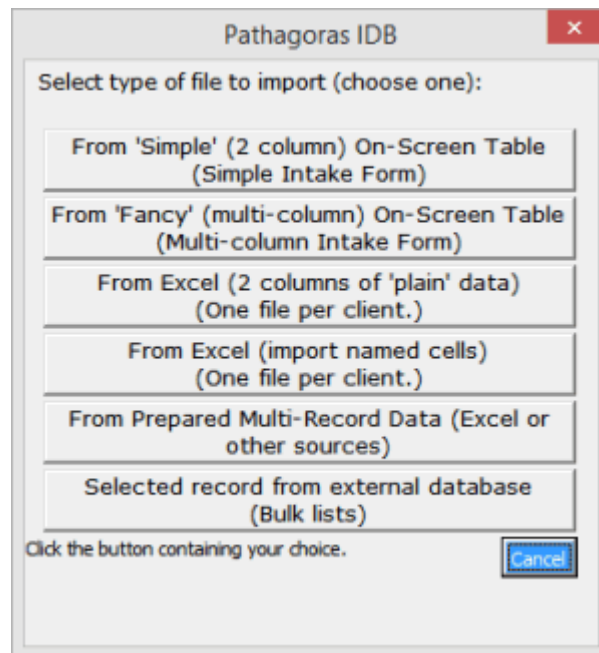
Scraping the Data:

Via the Instant Database screen.

Display the completed form (whether plain or fancy) onto the editing screen.

Review and edit as appropriate, remembering that the client/customer may have made typographical errors when completing the form.

Press Alt-D to call up the Instant Database screen. Then click the red Power Tools button and then the 'Import' button in the lower right quadrant. The following screen will appear:



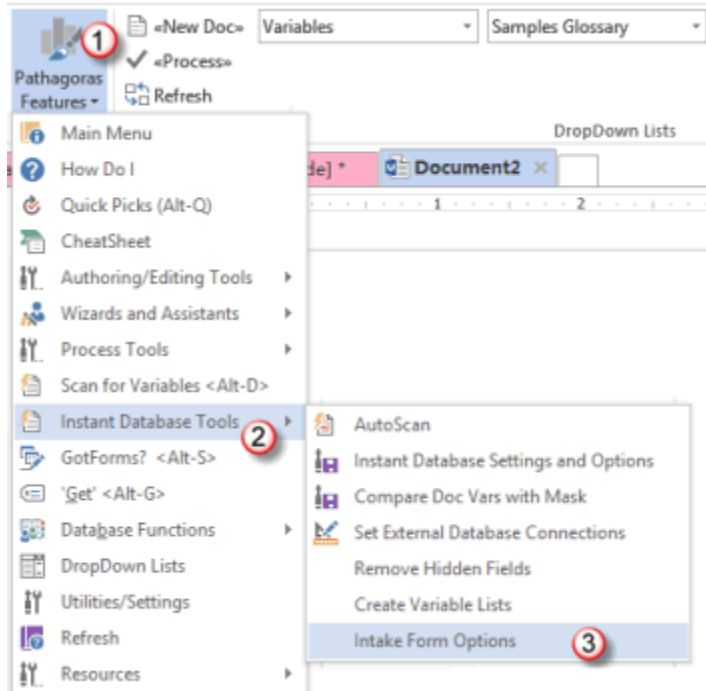
Click the appropriate choice from the top two buttons.

Via the Pathagoras Features menu:

Display the completed form onto the editing screen. (If the completed form arrived as an attachment to an email, open the attached file. If it arrived in the body of an email, copy and paste the table (or the entire email, it really doesn't matter to Pathagoras) into a Word document.)

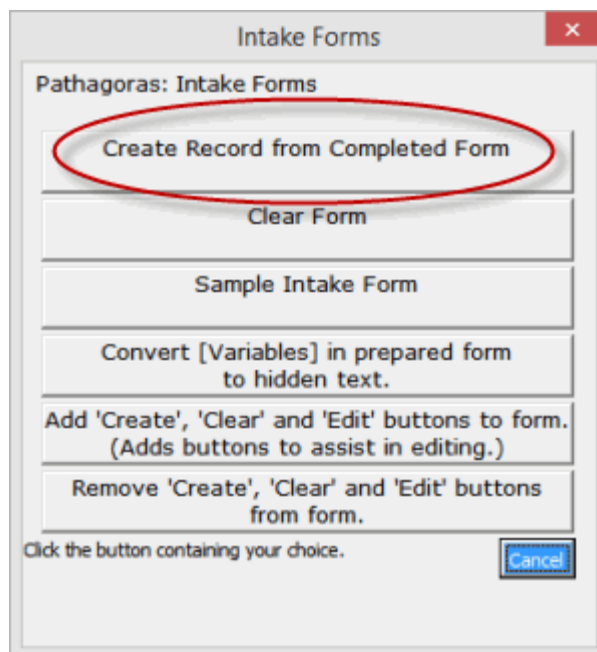
Review and edit as appropriate, remembering that the client/customer may have made typographical errors when completing the form.

Cascade down to the Intake Forms Options:



Click to collapse.

and select the Create Record (top) button:



Then save the record in an appropriate location. Then click the 'Create Record' button. Pathagoras will 'read' the form, including the hidden 'variable name' and the visible value, one cell at a time, and will save an Instant Database Record in the appropriate folder (where you otherwise save your IDB records). The next time you press Alt-D to bring up the Instant Database, the record will be present in the list of Existing Records.

➔ Don't forget: some clients are not good spellers. They don't always accurately remember dates. You still need to check behind the informant.

9.2.2.3 Transmitting the Form

- Send the form to the client (or other office worker) to complete.
 - Here you also have several choices.
 - You can send the form as an attachment. The end user would complete the form as a Word document and return it to you as an attachment to a reply email.
 - You can copy the form and paste it into the body of an email. The hidden text font will likely 'transfer' from you to the end user and back to you even though Word is not the transport medium. You may have to experiment with this. Preservation of the hidden text is essential



NOTE: If you send the form as the body of the email, the email setting must be RichText or HTML. It cannot be plain text.

Caveat: The hidden text essential to the form may not be 'seen' by all mail handling protocols. Be sure to test this out on your local systems. Even if the hidden text is not well handled, you can still send the client or customer the form which boldly displays the variable name (instead of hiding it). It's a choice of 'beauty' vs. 'function.'

9.2.2.4 Sample 'Fancy' Forms

You can download sample forms from these links:

This is a 'classic' style domestic relations intake form. Multiple items are requested per each row.

www.pathagoras.com/forms/Sample%20Form.docx:

(INSERT YOUR LETTERHEAD, LOGO, ETC. HERE)			
INTAKE FORM			
Your Name (First Middle Last): John Doe			
Address: 123 Main			
City: Newport News	State:	Zip:	Phone:
County:	How long lived in County:	How long lived in State:	
Cell Phone:	Email address:		
Date of Birth:	Birthplace:	Social Security #:	
SPOUSE INFORMATION			
Spouse's Name (First, Middle, Last):			
Address:			
City:	State:	ZIP:	Phone:
County:	How long lived in County:	How long lived in State:	
Date of Birth:	Birthplace:	Social Security #:	
MARRIAGE INFORMATION			
Place of Marriage (City, County, State):		Date of Marriage (Month, Day, Year):	
Wife's Maiden Name:		Number of Children Under 18 in Household:	
CHILDREN			
Name(First, Middle, Last):		<input type="checkbox"/> Male	<input type="checkbox"/> Female

This next item is a form that has a much 'sectional' design. It has multiple tables for the different aspects of the desired data. It is as much a summary of a transaction as it is a data collection form, but it is obvious how both purposes are met with the single form.

www.pathagoras.com/forms/Transaction%20Data.docx:

Details Of Purchaser	
Full Name :	
Short Name:	
Address:	
Telephone. No.:	
Fax No.:	
Email:	
Authorized Signatory:	
Signatory Title:	

Details Of Supplier	
Full Name:	
Short Name:	
Address:	
Telephone No. :	
Fax No.:	
Email :	
Authorised Signatory [:	
Signatory Title:	

Pricing	
FOB/CiF/Ex-works:	
Shipping Destination:-	
Country: India	

Insurance to be taken by Supplier	
Public Liability Insurance Amount in USD:	
Product Liability Insurance Amount in USD:	

www.pathagoras.com/forms/InputForm.Sample.docm. (This is a 'self contained' form. A bit experimental, with conversion buttons and instructions on the face of the form. More re: this kind of form will be posted in the near future.)

9.3 Excel Intake Forms

You can also use Excel as the tool to request data from your client/customer and to quickly and automatically import it into an Instant Database record.

This technique is discussed fully at the page entitled "[Excel and the Instant Database](#)".

9.4 Adobe™ Acrobat™ as Intake Form

A client or customer who has filled out an Adobe™ Acrobat™ PDF has provided all you need to create a new Instant Database record. Read more at this page.

The Pathagoras System

External Database Connections

Part



X

10 External Database Connections

Perhaps you have a collection of 'data' (contact names, addresses, etc.) in an existing database. And perhaps you would like to take advantage of that existing data to avoid as much as possible having to retype the existing information into the Instant Database module.

Pathagoras offers you several options for accessing that data. Read on.

10.1 Definitions and Concepts

External Database: An external database is any database that is not created by Pathagoras' Instant Database module. Common databases in contact record managers (CRMs) such as Act®, time and billing programs such as TimeMatters® and Tabs®, practice management systems such as PracticeMaster® and , as well as a wide variety of non-specialty database and similar programs, such as Excel, Access, etc.)

Links: For an on screen document that contains variables to use information from an external source to complete those variables, you must create a link to pair the on-screen document to the data source. Typically (i.e., in Word, without Pathagoras, you would use Word's mail merge functions to create that link. In every program other than Pathagoras, you would have to embed fields referencing the external database into your documents.

Sometimes the link you made to the database would be permanent. Whenever you recalled a document with a permanent link, Word would interpose a 'confirmation' that it was about to link to the database, slowing your startup. Sometimes the links would be temporary, but then in order to personalize your on screen document, you had to relink to the database, and sometimes you could not recall where the database was.

Fields: Those infernal 'grey-backgrounded' pieces of text that you typically must create within a document that you want to link to an external database. No fields of this nature are required with Pathagoras. Plain text will work.

Pathagoras has always prided itself on its plain text underpinnings. There are no complicated or hidden, fields to create, no confusingly constructed links to make your life miserable. Only plain text. You type and see what you want. The less confused you and the other end-users in your office are, the more productive you will be.

Pathagoras can offer you quick access to up to 3 external databases. While you have to know a little bit about your external database programs in order to make the initial call on it (and you may have to call upon your IT person to help you out this one time), once the connection is made the rest is simple, straightforward and intuitive.

Pathagoras can work with the following external database types:

.csv files: This may be a new term, but it is not a complicated concept. ".csv" means 'comma separated value' and refers to the way the data in your primary database is exported to a file. These are sometimes called "comma delimited" files. It means that the values contained in the file are stored in individual rows, one per record, with the breaks between each field of data in each row indicated by a comma. Here is what a comma separated file might look like:

```
Jane,Doe,1234 Main Street,Jacksonville,FL,32211,red
Harry,Johnson,914 Oak Tree Lane,Ft. Collins,CO,78765,white
Roy,Lasris,117 Chisman Landing,Seaford,VA,23696,blue
...
```

As you can see, The various elements of information (name, address, quantity, price, etc.) are separated from the other elements by a comma. Hence the name 'comma separated values. Typically the file extension of a .csv file is '.csv' (hence its common name) but some programs export to '.txt' files and others to '.dat' files. So long as the data is truly 'comma delimited,' you are good to go.

In many cases, the first row of the .csv file contains the names of the fields. This is for reference purposes. Typically refers to the field names assigned by the original database from which the file was exported. And these field names are used by the importing program to pair up fields if the data doesn't match the order or the spelling of the importing routine.

```
FirstName, Lastname, Street, City, State, ZIP Code, Color Choice
Jane,Doe,1234 Main Street,Jacksonville,FL,32211,red
Harry,Johnson,914 Oak Tree Lane, Ft. Collins,CO,78765,white
Roy,Lasris,117 Chisman Landing,Seaford,VA,23696,blue
...
```

Sometimes, the information itself contains a comma as a proper punctuation mark. Addresses contain comma between the city and the state, and if the field name is "Address" instead of the component parts "City", "State" and "Zip", an enclosing set of quotation marks are added so that the punctuation is not seen as a separating comma.

```
FirstName, Lastname, Address, Color Choice
Jane,Doe,"1234 Main Street,Jacksonville,FL,32211",red
Harry,Johnson,"914 Oak Tree Lane, Ft. Collins, CO 78765",white
Roy,Lasris,"117 Chisman Landing, Seaford, VA 23696",blue
...
```

.tsv files. Identical to 'csv, except the separator is a **tab**. Less frequently used, but still common nevertheless.

Excel files: Each row represents a record, and each cell in the row a value. The first row may contain the field names

Outlook contacts: Pathagoras can read and return selected fields from your Outlook contacts.

SQLServer and MySql databases.

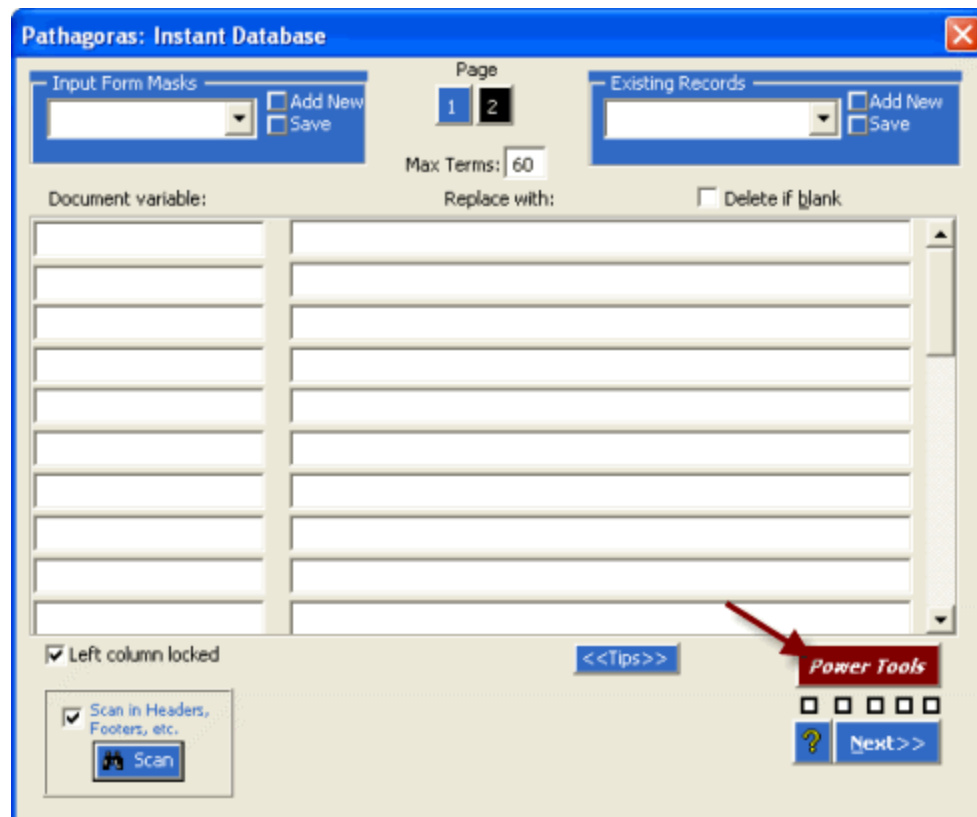
It is simply not possible (presently, anyway) to effect direct links to every external database, but since every database can export to .csv and .tsv and probably Excel, an 'almost' direct link is entirely possible.

10.2 Getting started . . .

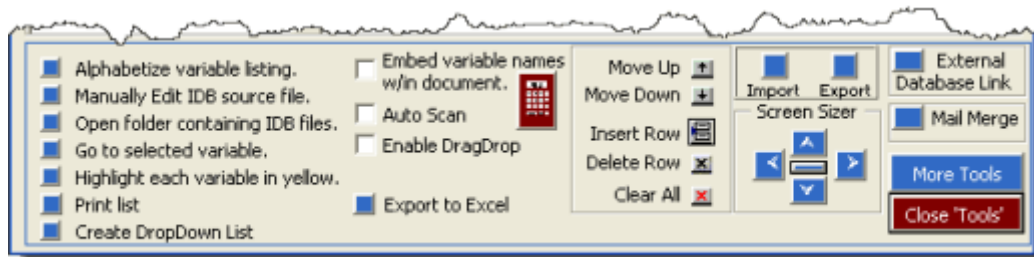
Pointing Pathagoras to your external database.

To draw information from your external databases, you will need to point Pathagoras to it. Here are the universal initial steps:

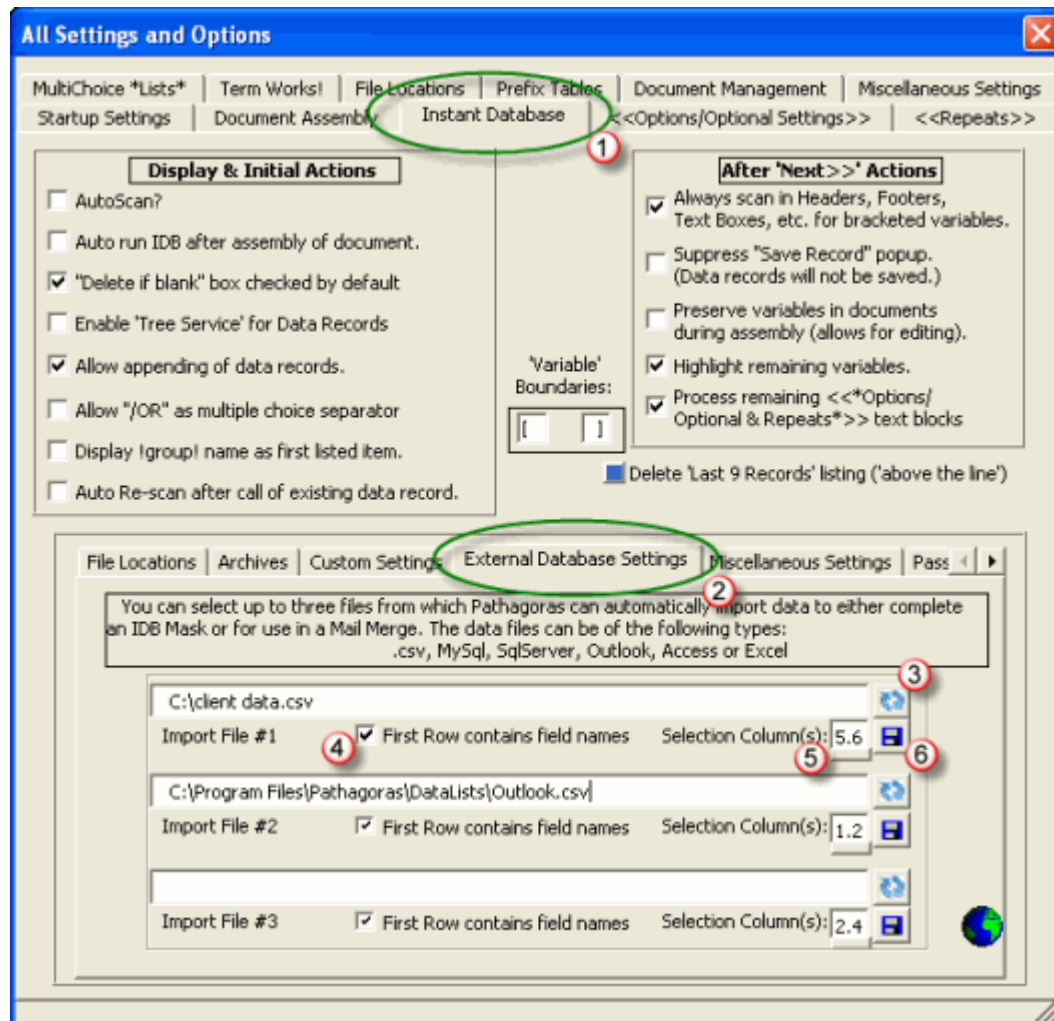
1. Display the Instant Database screen by pressing Alt-D. Click the Power Tools button.



2. The bottom section of the Instant Database screen repaints. See image below. Click More Tools.



3. A new screen appears, with the Instant Database Settings front and center. Click the External Database tab in the lower set of tabs.



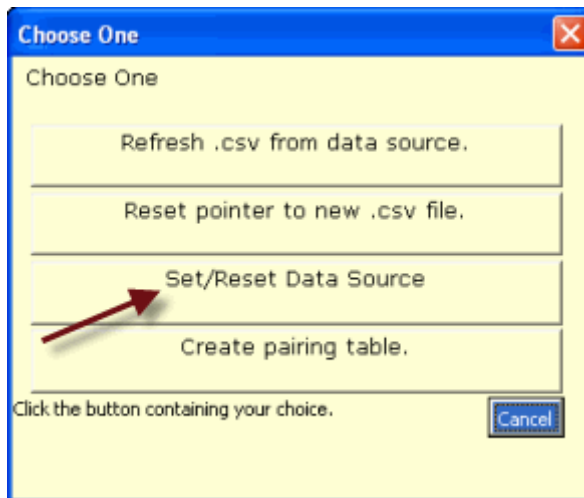
4. With the Instant Database tab (1) selected, click the External Database Content tab (2) in the lower tab set. Using the navigation tool, navigate to and select to the desired .csv or Excel database.
5. IF YOUR DATABASE IS YOUR OUTLOOK CONTACTS, FOLLOW [THESE ADDITIONAL STEPS](#) ²⁹⁰.
6. IF YOUR DATABASE IS A SQLSERVER OR MYSQL, FOLLOW [THESE ADDITIONAL STEPS](#) ²⁹¹.
7. If the first row of the database contains the field names, check the box so named (4).
8. When Pathagoras displays the data from the selected database, it will display up to two columns of data from the database so that you can make your selection. You can select which column(s) will be displayed. (Typically you would want to display a file name, or a last name and a first name, or a file number, something that will insure that you select the proper unique record.) Type in the Selection Column box (5) the number representing the field name or column number which contains the value(s) you want displayed. Separate the columns (if 2 are selected) by a comma. Because of the way the selection screen is designed, make sure that the 'longer' column is shown first
Click the button just beneath the "Selection Column" box to display the first 10 fields

of the database. That should assist you in making your display selection

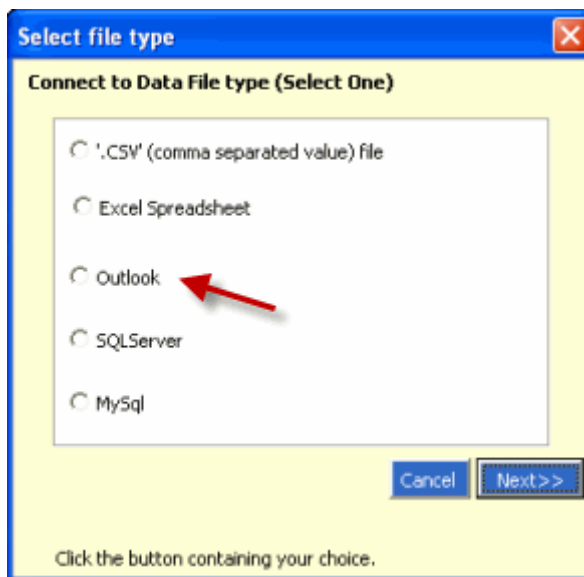
9. Repeat the steps in paragraph 4 above for a second and third external database files, if desired.

10.3 . . . with Outlook

When you click the



After you select from . . . you will see this screen.



Choose the Outlook option and click Next. This screen will appear.

Make such settings and selections as are appropriate. Click the Next button and your Contacts folder will be added as one of the three possible datasources from which you can draw values into the Instant Database system.

10.4 . . . SQLServer and MySql

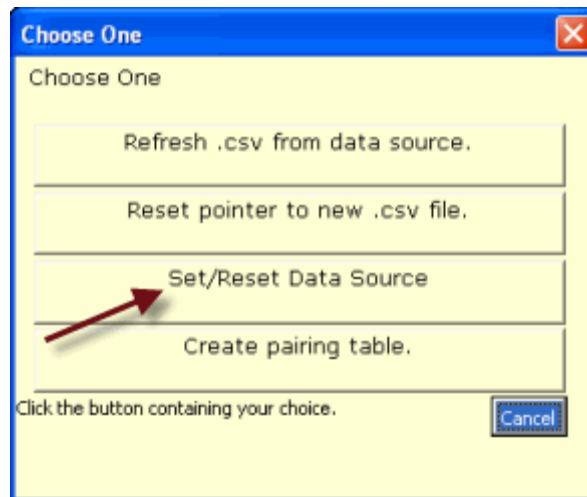
SqlServer or MySql are two database engines that enjoy immense popularity. They are becoming industry standards. Many proprietary databases use one or the other of these systems even without naming them specifically. (For example, TimeMatters(r) is based in SqlServer.)

Pathagoras can link to data stored in your SqlServer or MySql databases, and download that data (or, preferably, a sub-set thereof) to a local .csv file. so That csv file can in turn be readily used by Pathagoras.

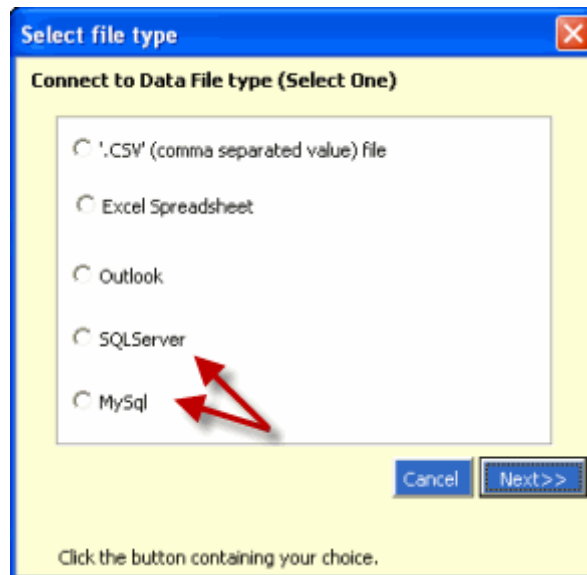
If you want to connect to SqlServer or MySql data, you will first need to tell Pathagoras how to connect to it. These databases are typically quite sophisticated and secure. Login 'credentials' are always needed to access the data.

Here are the additional steps needed to make the connections. (While the steps are somewhat numerous, you need only follow them one time for each table in a database to which you wish to make a link.)

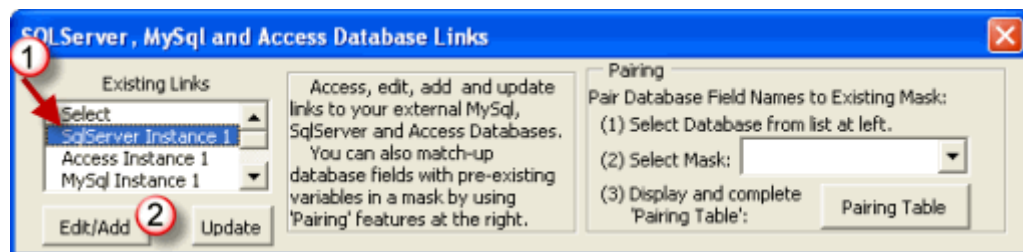
The preliminary steps are outlined on the [page in this Manual](#)²⁸⁷. They took you to this page. Click the Set/Reset Data Source button.



You will next see this screen.



Click the SQLServer or MySQL, as appropriate. This screen will appear.



Again, select the type of database (SqlServer, Access or MySQL) as appropriate (#1). Then click the Edit/Add button (#2). The following screen will display asking for your credentials and other linking information.

Pathagoras Database

Form Instructions

Local Name: SqlServer Instance 1

DB Server Name: 999.999.999.999 (1)

DB User Name: myname1

DB Password: [REDACTED] (Remember)

DB Name: path_sampledb

DB Port No.: 3306

Table Name: mytable

Display Fields (2)

Display Data (3)

New Instance: [REDACTED] (Create New, Rename)

Select All Select None

Sort By (4)

Download Data (5)

Fill in the appropriate login credentials (#1). Up to 6 fields will have to be completed before you can establish a connection to the database. (This is not a Pathagoras requirement. It is security requirement imposed by your database. It is a good thing.)

Note that the credentials will be different depending upon if the database is 'local' (your computer or local network) or Internet/cloud based. Consult with your IT department if you do not know your login credentials.

Once you have completed the required fields, click the 'Display Fields' button (2). If you have properly entered the credentials needed to log into the database, you will see a listing of the available fields. Select those fields that you want to be made available to Pathagoras. (Note the "Select All" and "Select None" buttons which may help to speed the selection process.)



Many database fields are 'administrative' in nature or not really applicable to the document assembly tasks at hand. If downloaded, these fields will only serve to slow you down as useless (to Pathagoras) information is downloaded and processed. You should take the time to select only those fields that you really need.

Next, click the Display Data button (3). All of the data in the selected fields will be displayed in the box in the lower right quadrant of the screen. You can view, scroll and sort the view to make sure that you have the information you are seeking. You cannot change the data. Pathagoras link to your data is 'read only.'

If you want the data presorted for more efficient use by Pathagoras, type the appropriate field name in the 'Sort by' box (4)

When you are satisfied with your selection, click the Download Data button. You will be asked to provide a name for the file. Note that the data will be written to a comma separated value

(.csv) file. Pathagoras must segregate the copied data from the source. Remember the file name. You will need to navigate to this file when you return to Pathagoras.

The screenshot shows the 'Pathagoras Database' application window. It has a 'Form' tab and an 'Instructions' tab. The 'Form' tab contains several input fields: 'Local Name' (SqlServer Instance 1), 'DB Server Name' (999.999.999), 'DB User Name' (myname), 'DB Password' (masked), 'DB Name' (path_sampledb), 'DB Port No.' (3306), and 'Table Name' (mytable). There is a 'Remember' checkbox and a 'New Instance' section with 'Create New' and 'Rename' buttons. Below the form are 'Display Fields' and 'Display Data' buttons. A list of fields is shown on the left, including FILENO, DATE_OPEN, CASE_TYPE, CASE_GROUP, CLIENT_LAS, CLIENT_FIR, CLIENT_MI, CL_FULLNAME, CL_SEX, CL_ADD1, CL_ADD2, CL_CITY, CL_ST, and CL_ZIP. A data table is displayed on the right with columns: FILENO, CASE_GROUP, CLIENT_LAS, CLIENT_FIR, and CLIENT_MI. The table contains several rows of data. At the bottom, there are 'Select All' and 'Select None' buttons, a 'Sort By' dropdown (set to client_las), and a 'Download Data' button.

FILENO	CASE_GROUP	CLIENT_LAS	CLIENT_FIR	CLIENT_MI
26896	WT	Nusbaum	Diane	M
24924	BK	Salang	Salbert	J
24007	BK	Pryor	Shauna	
24045	BK	Cascante	Jaime	R
24048	BK	Montgomery	Felicia	G
24055	BK	Little	Billy	
24056	BK	Muckelroy	Kenneth	G
24012	BK	Fowler	Tod	A

If you want to link to a second (or third, fourth, etc.) table, you should save out the current setup under an appropriate name. To do so, type the name in the 'New Instance' text box and click 'Rename'. Then you can make other selections, save out another instance, etc. There is no limit as to the number of links you can create.

10.5 ... with Excel

Pathagoras has strong connectivity with Word's sister program Excel. Using the tools discussed above, you can easily assign your favorite Excel spreadsheet which currently stores your company or firm's data to one of your three instant access databases.

- Information on how to pull Excel data directly into your Word document is found beginning at [this page of the Manual](#) ⁶¹⁴.
- Further, Pathagoras offers an Excel version of its program. Using that product, you can create a form in Excel designed to capture data that you can use both in Excel and (immediately) in Word. The benefit of the Excel method is that much fancier intake forms can be designed. The form can be completed locally or sent via email or posted on the web for complete by a client or customer. When the completed form is sent back, an Instant Database record can be created in three mouse clicks. Click the link to view the [Pathagoras for Excel Manual](#). See in particular Section 6 "Data Collection Forms" for instructions and illustrated examples.

10.6 Simple Merge of External Data

So, by now you have pointed Pathagoras to one or two external databases. Now, let's see what we can do with that data.

The first thing you need is a document with bracketed variables. So bring one up to the editing screen via the Document Assembly routine or via a DropDown list.

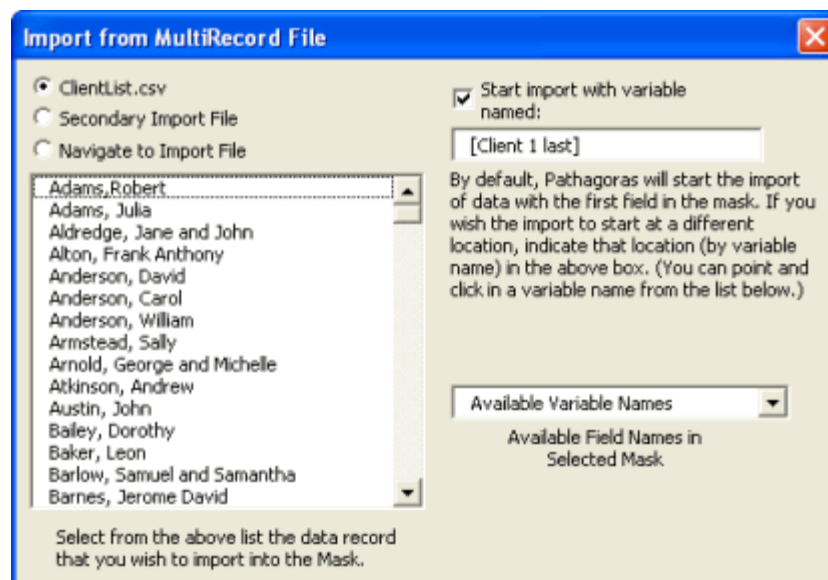
Next, call up the Instant Database screen (typically the press of <Alt-D> displays the screen). At this point, you have two choices:

Using external data with an Instant Database mask: If you have prepared a 'mask' which contains the document variables, call it up from the list in the upper left corner of the screen. (If the first row of your external database does not contain field names, you must use a mask. That is the only way that Pathagoras can know what variables your document contains.)

(If the first row of your external database does contain field names, use of the mask is optional. If you choose a mask in this case, Pathagoras will attempt to match the variable names in the mask with the database field names.)

Using external data with no IDB mask: If you do not pre-display a mask, then the first row of the external database must contain your field names. (Brackets are not mandatory. Pathagoras can add them, but the field names from the database must otherwise match the variable names in the on-screen document. Because both sides of the IDB screen are 'blank,' Pathagoras will complete both sides for you by placing the database's field names (with brackets added, if needed) in the left side of the IDB screen and the 'personal' values at the right.

Once you have decided on the above issue (mask vs. no mask), click the red Power Tools button and then click the 'External Database Link' button in the lower right quadrant of the repainted Power Tools section. A screen similar to the following will appear.



Select the database from which you want the data drawn. The first database is the default and the screen will be pre-populated with values from that database. Select the secondary file or the "Navigate to" option to display other choices. Select the record you want to use.

Once the IDB screen is populated with the external data, you can make any changes that you wish. We recommend at this stage that you click the Scan button to add any variables in the document that do not exist in the external record. Provide values for those newly added variables. (The variables that were added via Scan will appear at the bottom of the IDB screen, or on subsequent pages.) Click the Next button. As always you will be asked whether you wish to save the data as a new IDB record. Your answer depends upon how you intend to use the record in the future. If you are pretty sure you won't then don't save it. If you think you might, then save it.

Notes:

- While the primary import file always displays by default, you have instant access to the secondary import file (if set).
- You can navigate to any other data containing file, even if not registered.
- You can use the controls at the right to designate the mask variable you wish to start import into. (Just as a reminder: You can reserve the first X number of fields in your mask for keyboard input, and take advantage of the external data to fill in the rest, or as much of 'the rest' as exists in your import file. Just select from the controls at the right the first variable in the mask you wish to begin receiving the imported data.)

10.7 Updating Your Data

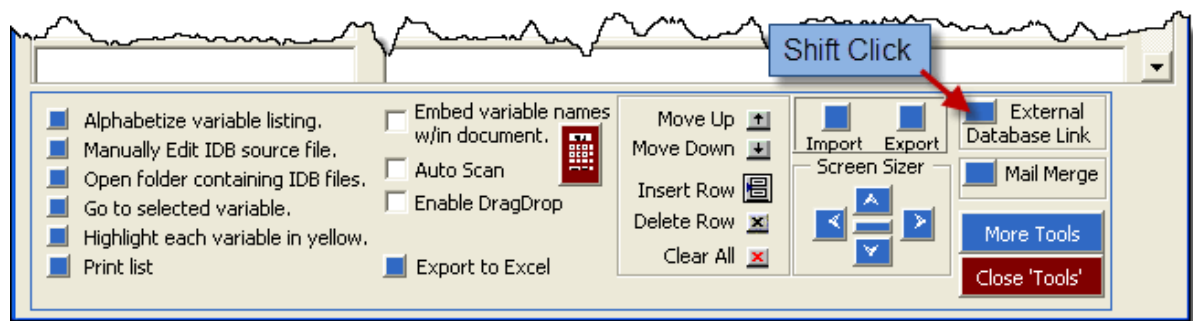
Over time your external data will change (new records, corrections to old records, deletion of obsolete records, etc.). You will want to update the local data to reflect those changes.

One user understandably said "I sure hope I don't have to manually link to my Sql data file each time I need to update my local data!"

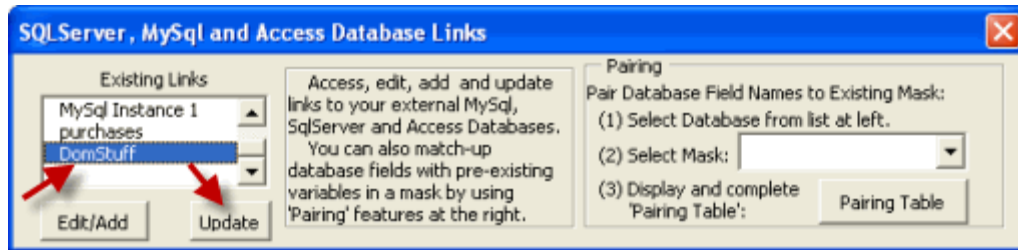
Don't worry, you don't. First of all, the credentials and selections are stored so you would never have to reenter them.

But Pathagoras offers more than that. Pathagoras offers a 'quick update' routine that avoids a return to the Sql Setup screen altogether. When you want to update your local data,

- (1) activate the Instant Database form (Alt-D) and click the Power Tools button (as before).
- (2) Shift-click on the External Database Link button.



You will be taken to the SQL Link screen shown below.



(3) Select the database you want to update and then

(4) Click the update button. Your data will be quickly updated.

Once your data has been updated, you will be returned to the Instant Database screen. Click the External Database Link button (no shift this time) and you will be able to select the record you wish to import.

10.8 Create Record via Web-based Intake Form

Technology being what it now is, you can design and post an Intake Form on your website which your client or customer can fill out and 'submit' to you. If set up correctly, the data will pair perfectly with your Instant Database system. Theoretically, the data submitted can also be used by your other office systems (billing, case management, etc.) so that you conceivably don't have to enter data even once.

Check out our [Design-a-Form](#) (Intake Form) page for more information on this.

10.9 Using *MultiChoice* Lists

A MultiChoice *List* is a collection of a series of values from which the user may choose. It is referred to in the source document by an alias that you assign when the list is created. So, the 50 United States can be individually listed in the MultiChoice *List* dictionary under the alias "States" and referenced in the source document simply as [*States*] (if a variable is desired) or <<*Options**States*>> if an options block is desired.

An entire section under the MultiChoice Lists portion of this Manual is devoted to linking to External Data. [Read more here.](#)⁴⁷⁹

10.10 Create Record via External Program

You can create individual Instant Database records that are perfectly usable by Pathagoras' Instant Database system using any program (PC or web based) that can generate a CSV (comma separated value) file.

First, a bit of super easy 'technical' stuff:

- Pathagoras stores the individual Instant Database records as separate .csv files. These files are stored in a single folder.
- When you initially install Pathagoras, the IDB folder is located as "C:\Program Files\Pathagoras\IDBS". (You can move the location of the folder. If you have done so, and forgot where you put it, click [here for instructions](#)²⁵⁰ on how to determine its location)

- When you press Alt-D (or otherwise call up the Instant Database screen), Pathagoras will read the names of each ".csv" file then present in the IDB folder. It then sorts the names alphabetically and pours the alphabetized list into the 'Existing Records' drop down list in the upper right corner of the screen.
- The 'read' of the .csv files into Pathagoras' memory is 'dynamic.' That is, every time you call up Instant Database, the folder is read anew and any new records that were placed into the folder since your last 'read' will be shown without any additional work on your part.
- The consequence of the last bullet point is that if you manually add a .csv file to the list, it too will be seen by Pathagoras the next time you display the Instant Database screen. So, if you can cajole your external program to create a single record's worth of information into a .csv file into the proper folder, that new .csv file will be displayed along with the other .csv files.
- Likewise, if you have an Intake Form creator (perhaps you have purchased one of ours or have designed your own Intake Form) that creates a .csv file in the style outlined below, you have all you need to add records to your system. Click the link to find out more about our [Design-a-Form](#)²⁹⁷ (Intake Form) services.

All third party programs have the means to export a single record into a .csv file. Your 'job' is to figure out how to do that, and then to just do it. You will need to consult your program's documentation to figure out the steps. Typically, it will be an operation called Export and typically that operation can be found under the "File" menu.

A few more points and requirements:

- The exported record must be in 'comma separated value' (.csv) format. As you follow the steps to export the record from your third party database program, this invariably will be an option that is presented to you.
- If offered, you should select "Quote comma" separation. If not offered, don't panic. Pathagoras can handle the conversion when it sees and begins to import the record.
- The exported record must consist of two lines and only two lines.
 1. **The first line of the record must contain the field names.** For what we hope are obvious reasons, these field names must be identical to names you have assigned to the variables in your source documents. So, if your external database has field names of "Client Name", "Spouse Name", etc. your 'Pathagorized' document variables must be [Client Name]" and "[Spouse Name]" If it is something else, there can be no matching up.

(Note: If, while exporting the data, your external database can add the square brackets that denote 'variable' within Pathagoras, great! But if your program cannot add the brackets, don't worry. Pathagoras will add them for you.)

2. **The second line of the record must contain the 'personal' values** that will be substituted for the variable at document assembly time. These values must be in the same order as the field names of the first line. (This ordering, including necessary commas to indicate 'blanks,' is handled automatically by the export program of your external database. So while the rule is essential, compliance is actually automatic. Just don't 'mess' with the result in a way that might violate the rule.)

10.11 Pairing Table

It happens with great frequency that the field name of an external database does not match the variable name you want to use in your document. For example, you want [Testator Name] and [Spouse Name] in your Wills, but your database gives you only [Client1 Name] and [Client 2 Name]. Of course, you can readily re-write your Wills to accommodate those names, but Pathagoras offers another choice: a pairing table whereby you can tell the system that you want to pair [Testator Name] with [Client1 Name], [Spouse Name] with [Client 2 Name], etc.

To create the Pairing table, navigate to the External Database Settings tab under the Instant Database Settings and Options screen. Click the 'Set' button (the one with the circling blue arrows)

10.12 'Fielded' Database Connections

Using 'Mail Merge' links provided in Word:

While Pathagoras 'essence' is avoiding fields, Pathagoras actually works quite well with source documents that are linked (using 'merge fields') to external databases. The actual process of linking your source documents to your database(s), however, is covered in a separate publication for advanced users. Check out [External Database Connections Manual](#) (pdf file)

For what it is worth, we have discovered by survey that linking to external databases via Mail Merge is a rarely used feature. Part of it is due to the complexity of creating 'those dreaded fields' to the novice user.

Even among those who we know to be sophisticated uses of external databases, Pathagoras' IDB is still by far the tool of choice for database functions.

"Why is that?," you may ask. "Why would advanced users choose to put their data in twice?"

- The main answer is that the data collected by the external systems simply is not the same data needed for the documents being assembled in Pathagoras. (Some is, of course, but not sufficiently enough to make the effort at linking everything together worthwhile.)
- Take, for example, a Last Will & Testament. The external database might contain the names of the Will's maker and his or her spouse, but rarely would the external database contain the names of every beneficiary or the Executor's or Guardian's names. And the 'Trust Termination Date?' Of course that wouldn't be there.
- Document specific variables can best be handled within the document.
- On the other hand, if the data stored in an external database is substantial and useful for completing your documents, and if the remaining variables needed to complete the document is insignificant by comparison, then it makes sense to use primarily the external data source. The Manual referenced above will take you step-by-step through the process.

11 Mail Merge

Do you have a need to mail the same letter (perhaps a 'Season's Greetings' letter) to multiple addressees. Do you want to create a series of envelopes to clients or customers in your Instant Database records? Perhaps you have a collections practice and need to send the same demand letter to multiple debtors.

That is what 'mail merge' is all about.

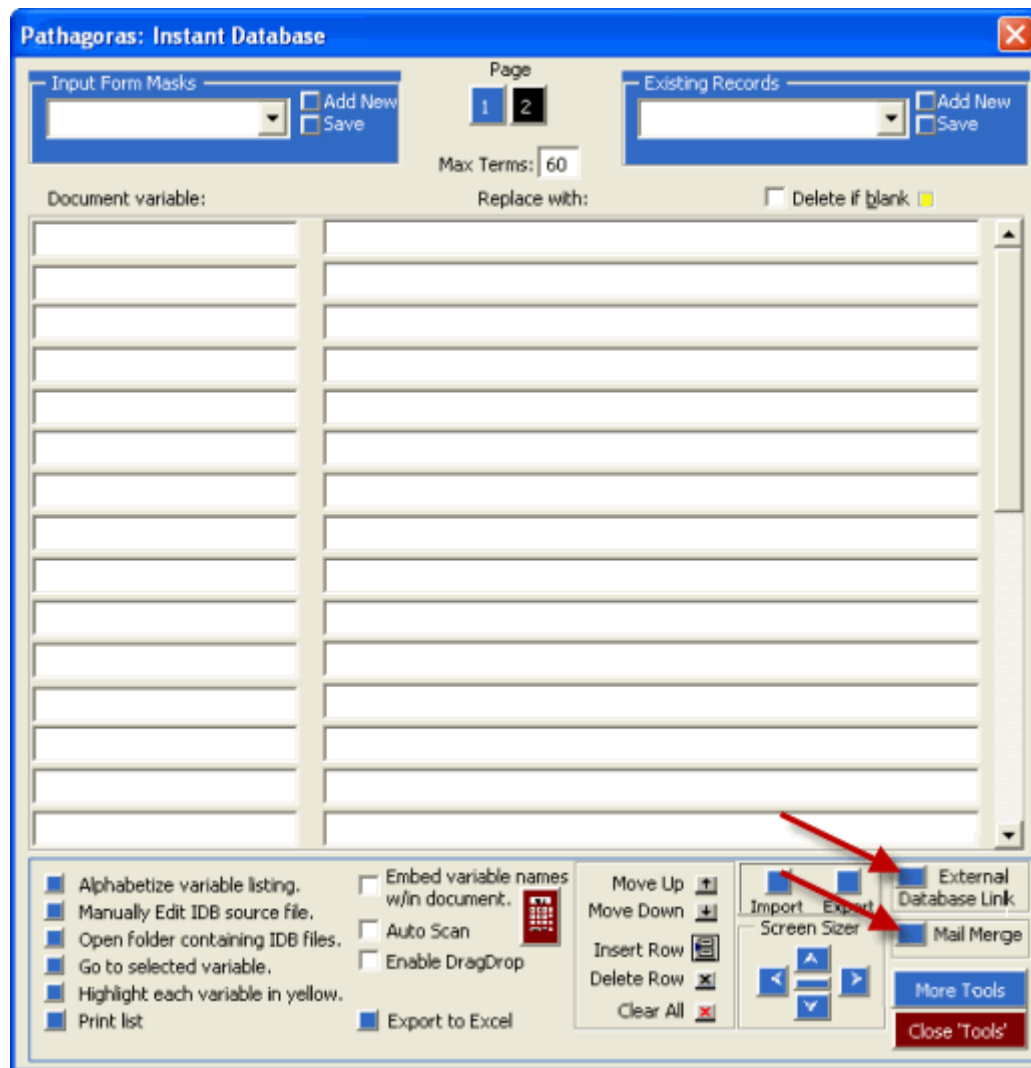
Pathagoras provides an easy way to accomplish your mail merge tasks, and it can all be done with plain text. (We believe that Pathagoras is the only program on the market to offer "plain text" mail merge. It is really quite a remarkable tool. The possible 'draw back' is speed, but the time difference is only a matter of seconds, so it's mostly a non-issue. Plus, the additional time needed to set up documents with fields vs. plain text likely will make up any difference.)

Your source of names can be from one of several sources. You can use your current Instant Database records. Just make sure that the variable names in your record marry up to the variable names in your mail merge document.

If you plan to draw names and addresses from an external data source, you must point Pathagoras to at least one external database in the External Database settings. See the [Setup section](#) ²⁸⁷ of this chapter. (You can set up to three external sources from which to draw on.) As with Instant Database source, you just have to make sure that the field names in your data record match up to the variable names in your mail merge document.

To begin a Mail Merge session, display on your editing screen the document you wish to use as the mail merge template. (The document can be an existing document that you recalled to the screen, or it can be a document that you just now assembled. So long as it has [bracketed variables] that will pair up with your data, you are ready to go.)

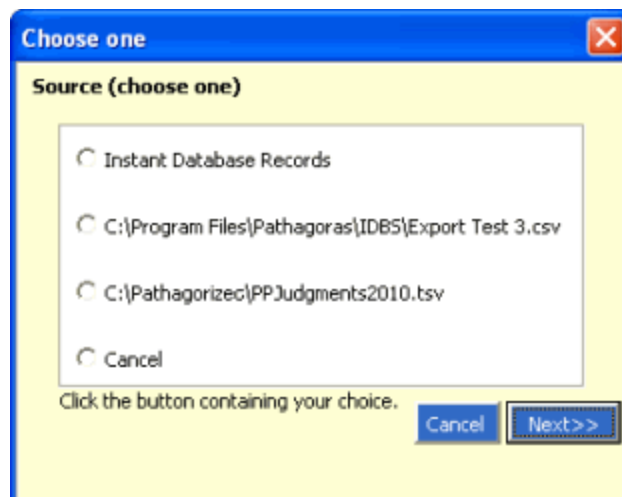
Next, press <Alt-D> to bring up the Instant Database screen. Click the Power Tools button to display the Power Tools options (image below).



If you have not previously set an external database link, do so first (the button indicated by the upper arrow).

If you have set the link, or are going to use your Instant Database records as your data source, click the Mail Merge button at the right. A screen will appear asking you confirm that the on-screen document is the proper mail merge letter.

Say "Yes" and the following screen will appear. Click the radio button next to the source of the Mail Merge data and then press Next.



Pathagoras quickly reads the data source and presents a checklist from which you can select the record(s) you desire. You can select a single record, all records, or any number in between. Regular list selection controls (click, ctrl-click, shift-click) will work. When you have completed your selections, click the Merge button. Pathagoras will generate one letter per selected record. The results are presented on screen for review.

Please note the following:

Variables in headers, footer and textboxes add processing time. If you are processing more than, let's say 30 records and you need header/footer data, you may wish to 'fake' the header/footer. Do so by changing the margins of your document and place the same text at the extreme upper or lower edges of your document. (If you have fewer than 30 records to process, don't worry about the above. It will still take longer to process, but it is only a matter of a couple of seconds. It becomes significantly noticeable when you have more than 100 records.)

Blank fields: You have a choice about where to delete blank fields. If you do not delete blank fields keep in mind that any fields that do remain will be 'called' the same thing. This works great when the remaining field is, let's say, a court date that will be the same for all records. Just scan the resulting mail-merged result, put in the date and all [Court Date] variables will be replaced with the same value. If you need different values for any of the remaining fields, perhaps mail merge was not the correct technique.

"Good" blank fields vs. "Bad" blank fields: What if, after the initial mail merge, you have variables such as [Court Date] (which are probably okay, since you want to assign them later) and also variables such as [Address Line 2] which are present simply because you chose not to delete empty variables in the initial mail merge?

Actually, IDB as a post-merge editing tool works quite well for this situation. After you have run the mail merge, call up the Instant Database screen. Click scan. You will see [Court Date] (the 'good' variable) and [Address Line 2] (the 'bad' one).

Complete the former with a date, and leave the latter blank. Make sure you check the "Delete if Blank" box at the top of the Instant Database screen.

When you click the Next button, [Court Date] will be replaced throughout the document with the date provided, and all instances of [Address Line 2] will be removed. (The assumption here is that since the initial mail merge process did not replace [Address Line 2] with a value, that no such value exists. Running a 'regular' IDB process call against the document will remove those variables if the "Delete if Blank" selection made.)

Calling in text from other documents: Unlike Word merges, Pathagoras can call in entire blocks of text into your merged document. It actually is a two step process, but when it runs, it is actually pretty seamless. The first step of any mail merge routine is the replacement of variables set out in a document with the values assigned to those variables in the data source for the mail merge. So typically [Client Name] will be replaced sequentially with each "Client Name" in the data (John Q. Doe, Robert R. Roe, etc.)

But with Pathagoras Mail Merging, it is not necessary to replace variables with just names and addresses. You can replace variables with calls to whole documents and snippets that Pathagoras recognizes. (Before making the most of this feature, you should fully understand that when Pathagoras sees <<document name>> in a document, it will hunt down the named document and insert its text in place of the call. See this [page of the Manual](#)¹⁰⁴ for more information.

So, if you set the value of a field not to a name like "John Q. Doe" but rather to a 'document call' (e.g., "<<document name>>") after Pathagoras runs the mail merge, it will immediately perform any 'document calls' that remain. For example, let's say that special text is needed in a mail merge document for employees over 55. In the data source, a field that you might call "AgeText", you would insert (or if using Excel or other database program, program it to work automatically if the employee is over 55) "<<Age55 Letter>>". Be sure to compose a letter called "Age55 Letter.doc" (or .docx) and save it in a folder where Pathagoras will find it. (See [Hunt Path](#)⁹⁵ for more info on that.) The actual flow is that as the mail merge is taking place, "<<Age55 Letter>>" is replacing [AgeText] as appropriate. At the conclusion of the merge, Pathagoras immediately begins to process the document, replacing <<Age55 Letter>> with the text of the document called "Age55 Text.doc". It's all pretty remarkable and very fast.

The Pathagoras System

Clauses: Adding and Editing

Part



XII

12 Clauses: Adding and Editing

Creating and Storing Clauses

A '**clause**' is the smallest component of a document assembly system. A clause is the building block of future documents. Properly constructed, they will allow you the infinite document assembly possibilities.

Pathagoras stores clauses in a way quite unique from its competitors:

- Pathagoras saves clauses in standard Word folders
 - Pathagoras saves clauses as standard Word text.
 - You can edit your clauses using ordinary Word editing techniques.
 - If Pathagoras were to 'disappear,' you could still find your clauses using ordinary Word navigation techniques.
 - The tools and shortcuts that Pathagoras provides are intended to augment, not replace, Word's features.
-

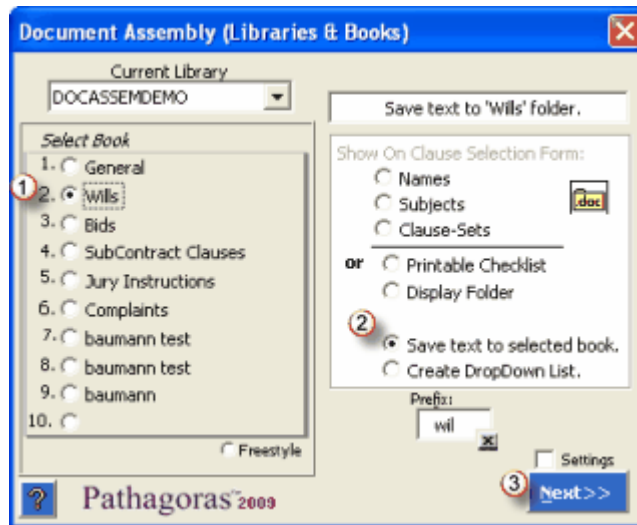
There are numerous (seven, to be exact) ways for you to add clauses to your document assembly books. A fuller discussion is found in the subparts that follow:

1. [Manually](#)³¹¹: Simply save a new document to the folder. It is no more complex than that!

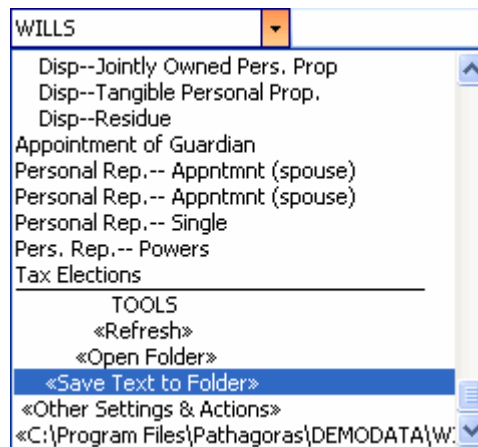
Explanation: Pathagoras performs most of its essential tasks by dynamically reading the content of existing document folders, and reporting what it finds. If you add (let's say) a 21st document to a folder used for document assembly, and then activate the Clause Selection Screen, all 21 documents will display. We don't think anything could be easier.

All of the more automated techniques 2 thru 7 below are variations of this theme. All that is happening when you invoke any of these processes is that Pathagoras will add a new document to the folder to which the book points. (The reason why you may wish to use one of the below techniques is to avoid having to remember the location of a particular book.)

2. [Via Libraries & Books Screen](#)⁴⁶: With the text you want to save on display, click the Document Assembly icon (which will display the Libraries & Books screen), select the appropriate Book and click the "Save to" button. Pathagoras displays the **Term Works!** screen where you can name the term and assign a subject. Click Next>> and you are done. It couldn't be easier.



3. [Via Highlight & Add](#)³³⁷. Highlight text from any source. Press <Alt-G>. The **Term Works!** screen will appear. Provide a Name and a Subject for the new term. Point to the book (folder or glossary) into which you want the term saved. Nothing more.
4. [Via a DropDown List](#)³⁵⁶. If a folder is displayed in a DropDown List, this method will be the easiest. Quickly add a term by highlighting the text and clicking the "Save Doc to Folder" element that appears at the bottom of the list.



5. [Via SaveSmart](#)⁵⁴⁷. While SaveSmart is a document management tool, it works quite well in supporting the document assembly module. You will want to assign a folder into which you wish to save new documents as one of your SmartPaths. Once you have done that, you can easily save any text (whether a complete document or just a portion of a document, into the assigned SmartPath.
6. Via Instant Book. Take a complete document and place simple characters -- we like to use '(*)' -- to mark each place a new clause begins. Then let Pathagoras automatically dis-assemble the marked up document it into its component pieces. Optional: After the disassembly is complete, create different versions of some or all of the new clauses to reflect different circumstances. You will soon end up with a substantial system from which an author can select among a wide variety of choices. This technique is more fully discussed and illustrated in the separate pamphlet [Document Disassembly](#).


7. Via Bulk Add: A variation of Instant Book. The initial mark-up is a bit more elaborate than Instant Book requires, but the results can provide a more functional book from the outset. This technique is more fully discussed and illustrated in the separate pamphlet [Document Disassembly](#).

Hopefully, as you read through this manual, you will see and understand that, with Pathagoras, adding a new clause to any book is as simple as composing text and saving that text into a Word folder. And editing the clause is nothing more than opening the document you saved and making the changes you want.

12.1 .doc vs. .dot

Pathagoras recommends that your source text (your boilerplate language and your assortment of alternative clauses) be stored in regular Word documents. That is, your source clauses should be stored as ".doc" (or ".docx" for 2007 and later) files, not .dot (or .dotx) files..

- .doc files are easier to save where you want them saved. (If you change a ".doc" to a ".dot" file, Word automatically changes the default save location to your templates folder. Then you must navigate back to the folder where you wanted the original stored. This folder switch does not occur with .doc file. Just save your substantive text in standard files. Store them where they make the most sense to you.
- .doc files are easier to access and edit. Typically, when you try to open a '.dot' file, Word assumes that you want to create a copy of the file, not its original. You have to remember and follow the steps to open the actual template file. Not so with '.doc' files.

 This ".dot" protection offered by Word make it more difficult to accidentally overwrite a template. However, you need not worry about overwriting originals during a Pathagoras session either, even with a 'doc' or 'docx' extension. Pathagoras always uses a copy of the original text when it assembles documents, never the original.

In all of the automatic clause creation methods discussed in the following sections, Pathagoras will always save the resulting clauses as .doc or .docx files.

12.2 Glossary vs. Folder of Terms


Other sections discuss and compare the benefits of storing your terms within glossaries vs. folders, and vice versa. Not knowing which path you are taking in this Manual to learn more about Pathagoras, we here offer a link read more about this important topic. See [Glossary vs. Folder](#)⁷⁵.

12.3 Via Libraries & Books screen

Perhaps the easiest way to add text to an existing book is via the Document Assembly (Libraries & Books) screen.

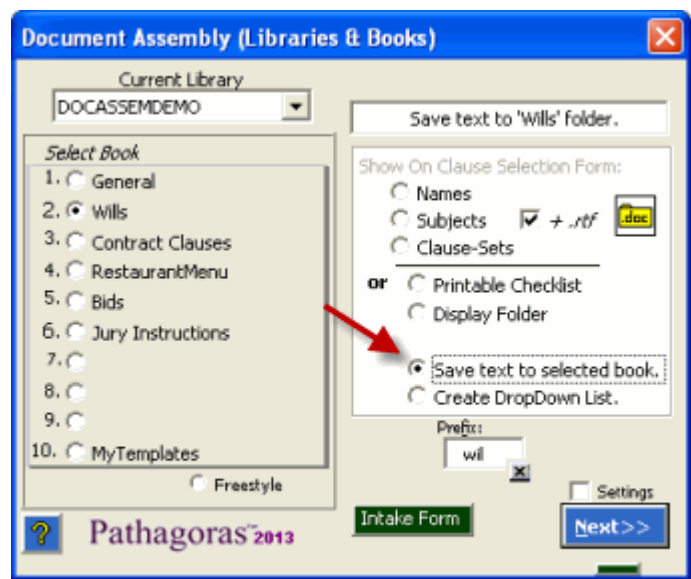
Here are the steps:

- (1) Display the text you want to add to the book.

 The text you choose to add to the book can be anything. It can be text, it can be a picture, a chart, etc. You can copy text/data/pictures/tables/links/whatever from

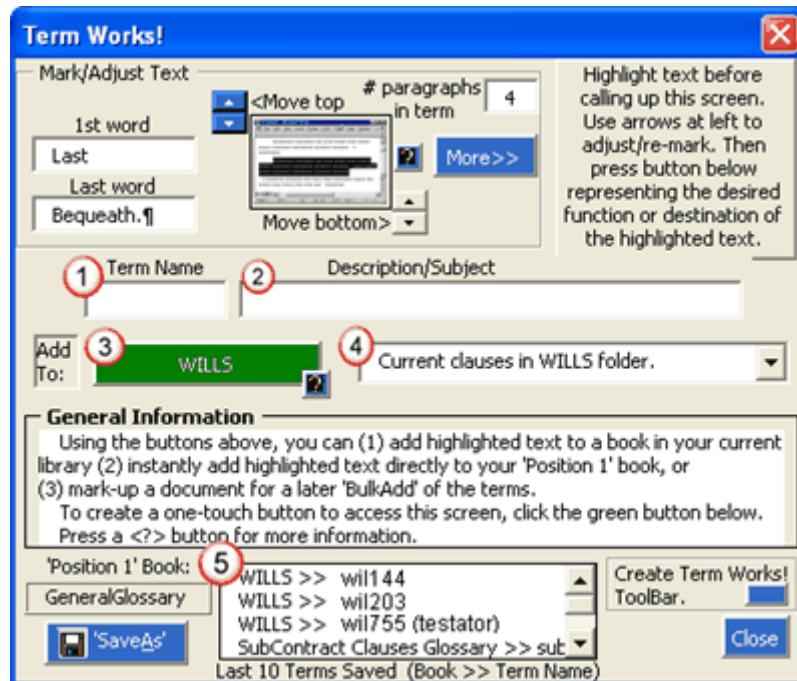
the Internet and paste it onto the editing screen. You may have text from a forms book that you want to add to your collection of forms. It does not matter to Pathagoras what the source of the text is. So long as you can display it onto your Word screen, it can be added to a book.

- (2) If you want to add just a portion of the displayed text, highlight that portion. If you want to add the entire document, don't highlight anything.
- (3) Press the Document Assembly icon. (It is the third one from the left, next to the Pathagoras features list.) The *Libraries & Books* screen will display.
- (4) Select the target book from the list of books at the left of the screen. If the book is not in your current library, select a new library from the drop down list at the right.
- (5) After you select the target book, the box in the middle will display a series of options. Choose "Save text to selected book." See the illustration below.



'Save text to selected book' selected.

- (6) Click the Next>> button. Pathagoras will display the *Term Works!* screen.



Note these features on the *Term Works!* screen:


- (1) & (2) A space to enter the new term's name & a subject.
- (3) the name of the selected book. This is also the "action" button that you will press to complete the 'add term' process.
- (4) a drop down list containing the names of all terms currently in the selected book. (This helps in consistency in naming, and to prevent duplication of names.)
- (5) a list of the 10 most recently added terms.

When you have named the new term and provided a subject, click the **Green Bar** (which will display the name of the book) to add the text to the book.

12.4 Via DropDown Lists

A block of text can also be quickly added to any folder or glossary that is reflected in a DropDown List.

Here are the steps:

- (1) Display the text you want to add to the folder or glossary.
 -  The text you choose to add to the book can be anything. It can be text, it can be a picture, a chart, etc. You can copy text/data/pictures/tables/links/whatever from the Internet and paste it onto the editing screen. You may have text from a forms book that you want to add to your collection of forms. It does not matter to Pathagoras what the source of the text is. So long as you can display it onto your Word screen, it can be added to a book.
- (2) If you want to add just a portion of the displayed text, highlight that portion. If you want to add the entire document, don't highlight anything.
- (3) Drop down the DropDown List.

(4) Select the entry "Add H'lighted Text". When the Term Works! screen appears, provide a name and subject for the clause. When you press OK, the clause is instantly added and will appear the next time you display the DropDown List. Further, if the folder or glossary represented by the DropDown List is also one of your books, the new clause will appear in the book the next time you display it via the Libraries & Books screen.

12.5 Term Works!

Term Works! is yet another method by which you can add text to your books. It is a powerful tool that you likely will use more often than any other because of its speed and ease of use. **Term Works!** is a collection of tools that makes it easy for you to:

- (1) add new terms and clauses to an existing book;
- (2) create a new book and add clauses to it; and
- (3) mark up text in preparation for a Bulk Add (document dis-assembly) routine;

Performing any of the above functions is a simple 'highlight and add' process. With **Term Works!**, you highlight the text (from any source) that you want to capture, and press <Alt-G> from the keyboard. Pathagoras can automate the rest.

An entire section of this Manual is devoted to the Term Works! function. [Click here to read more.](#)

334

12.6 Adding Clauses Manually

Pathagoras allows you to add clauses to your books in several different ways. Since they all lead to the same result (an augmented book of clauses), you should select the one with which you feel most comfortable and stick with it for a while. Don't try them all at the same time.

The 'manual add' technique you use will depend upon whether you are adding terms to a folder or a glossary.

Please note. Pathagoras is a very dynamic program. Regardless of how the new documents get into your folders or glossaries,

- when you initiate a document assembly session, Pathagoras will read the then current content of the specified folder (the 'book') and present whatever it finds onto the Clause Selection Screen.
- when you create a DropDown List, Pathagoras will read the then current content of the specified folder and pour whatever it finds into the List.

If you delete documents from the folder, the diminished contents of the folder will display the next time you perform Document Assembly or create (or refresh) the DropDown List.

12.6.1 To a folder of clauses

Adding terms to a *folder of terms*:

In addition to the more automated methods of adding content to your source folder discussed elsewhere, you can easily add new content *manually*. To add content to a Pathagoras book maintained as a "folder of terms," simply add more *documents* to the folder to which the book points.

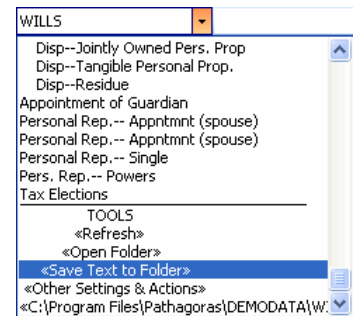
- You can take original content and save it to the folder.
- You can take a complete document and manually disassemble it into smaller components, saving each component as a separate document.

Via the Libraries & Books screen

Display the document you wish to add to a book. If you wish only to add a portion of the document (a phrase, a paragraph, a section, etc.) highlight that portion. Click the Document Assembly icon from the Pathagoras Toolbar. Select the book into which you want to add the text. Click the Save To option button from the center of the window. Press Next. The Term Works! screen will appear. Complete the name and subject boxes. Press the green button. That's it. (inset a picture.)

Via a [DropDown](#) List:

If the folder into which you wish to add a document is reflected in one of your DropDown Lists, it is a very simple matter of dropping down the List and clicking the "Save Doc to Folder" entry. Pathagoras will instantly save the document into the folder.



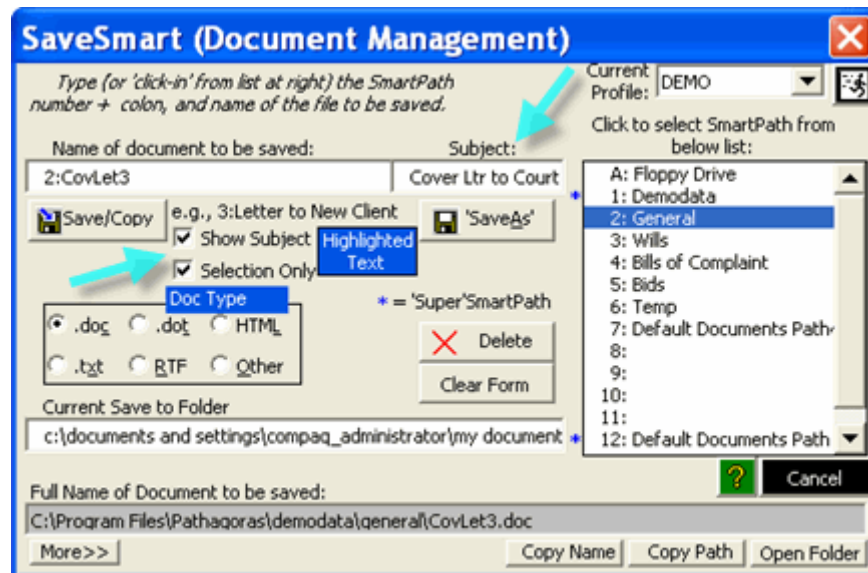
Via [Term Works!](#)

Term Works! allows you to 'highlight & add' the information you want into any folder in your current library. Just select (highlight) the text and press <Alt-G>. When the Term Works! screen appears, give the term a Name and a Subject. Select the target book from the list and Pathagoras handles the rest.

Via [SaveSmart](#)

Perhaps you want to save document to a folder that has not been assigned to a book. Let's further assume that you have no intention of creating a book with these clause. Pathagoras document management tool "SaveSmart" lets you highlight a block of text and quickly save it out as a new document. Here are the steps:

1. Highlight the text you want to save.
2. Click the SaveSmart icon. Because the text was highlighted, SaveSmart presumes that you want to do something 'special' with just the highlighted text. It augments the information normally shown on the screen. (Note in particular the 'Highlighted Text' block, and the addition of the text box labeled 'Subject' to the right of the Name box.



Using SaveSmart to create a new document from a highlighted portion of text.

3. Give the new document a name and a subject. If the folder into which you wish to save the document appears in the list of numbered folders at the right, click the appropriate folder and then press the 'Save/Copy' button. If desired folder does not appear in the list, click the 'SaveAs' button and you will be given the opportunity to navigate to the desired folder.

12.6.2 To a Glossary

Adding terms to a glossary:

From External Location:

If the source of the text that you want to move into a glossary is "external" to that glossary (a Word document that contains any text, images or tables, etc., from any source) use [Term Works!](#)³¹¹ Term Works! quite literally allows you to 'highlight & add' the information you want into any glossary in your current library. Just select (highlight) the text and press <Alt-G>. When the Term Works! screen appears, give the term a Name and a Subject. Select the target glossary from the list and Pathagoras handles the rest.

From Within Open Glossary:

If you already have a glossary open and displaying on your editing screen, and you want to add text while inside the glossary, here are two techniques to accomplish that:

1. **Use Term Works! (same as above).** Highlight the text you want to add (we presume the text is inside the actual glossary document, just not inside of bookmarks) and press <Alt-G>. Name the term and give it a subject. Select "This Glossary" from the list of targets and Pathagoras handles the rest.

--OR--

2. **Manually add your own bookmarks.** Highlight the text, click "Insert|Bookmark" from Word's menu area. Name the bookmark. On the lines immediately preceding the now

bookmarked text, type the name of the bookmark and then (on a separate line) the Subject you want assigned to the term. (Look at other glossary terms in the glossary for examples as to how the 'final product' should look. The red and blue colors identifying the term name and subject, respectively, are helpful, but not required.) See [The Technical Side to Glossaries](#)⁷¹ for more information.

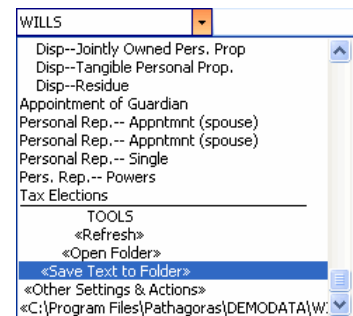
Other Techniques:

Via the Libraries & Books screen

Display the document you wish to add to a book. If you wish only to add a portion of the document (a phrase, a paragraph, a section, etc.) highlight that portion. Click the Document Assembly icon from the Pathagoras Toolbar. Select the book into which you want to add the text. Click the Save To option button from the center of the window. Press Next. The Term Works! screen will appear. Complete the name and subject boxes. Press the green button. That's it. (inset a picture.)

Via a [DropDown](#)³⁵⁶ List:

If the glossary into which you wish to add a document is reflected in one of your DropDown Lists, it is a very simple matter of dropping down the List and clicking the "Save Text to Glossary" entry. Pathagoras will instantly save the document into the folder.



12.6.3 The Quickest Glossary Ever

Okay, this admittedly is a hyperbolic name. Other techniques may in fact be easier and faster for you. The purpose of this page and the exercise it describes really is to demonstrate that there is nothing 'magical' about glossaries.

And there are circumstances where this indeed is the best approach.

Take an existing document. It should be a document that contains more or less 'standard' clauses that you want to convert to individual glossary terms.

Rename the document (using your SaveAs function) so that the document's name ends with the word "Glossary" (no quotes, and the case doesn't matter). With that simple document name change, you have created a Pathagoras glossary.

Now, of course, the glossary has no terms in it. But adding them is easy.

Highlight that portion of the text that you want to become your first glossary term. It can be a paragraph, several paragraphs, whatever. Press <Alt-G> against the highlighted text. The TermWorks screen will appear. You will see two boxes in the TermWorks screen for you to provide a Name and a Subject for this term. You must use a 'legal' bookmark name for the term's Name. A legal bookmark name is a single work that begins with a letter and contains no 'special' characters such as '\$', '%', '#', '@' etc. An underscore between words (like this: "Intake_Form") is fine.



We recommend that you name the term following the [prefix/suffix naming style](#)⁴²⁰. That

will help you to sort the order of the clauses in the Clause Selection and/or DropDown Lists with more precision. ('Names' of documents rarely fall in alpha-numeric order, do they?) Use the 'Subject' field to better describe the term. Subjects can easily be displayed, while the order of the names are preserved, but 'hidden' behind the scenes.

After you have given the term a Name and a Subject, press Green bar that says "This Glossary" and the glossary term will be created.

Keep on highlighting, <Alt-G>ing, naming and saving. You will have a complete glossary in no time.

12.6.3.1 Easy As Pi Glossary

Glossaries are merely an alternative form for storing text. In 'normal' Word/Windows use, you would click the Save button, give the document containing the text you want to preserve a name and designate a folder into which the document is to be stored.

With glossaries, you would still 'save' your work, but instead of saving your text in separate documents, you save it in a single document. Instead of 'documents' being the electronic storage bookends, you would save the text in the same document, using Word's bookmarks to indicate the begin and end points of the text. The name you assign to the bookmark is parallel to the name you assign to the document.

While there are more elaborate ways to create glossaries and to save text, here is our "Easy As Pi" method.

Create a two column table. The right column should contain the text you want to add. Typical content of this right column would be a signature block for the attorney, or an address block for a correspondent, but it could also be text snippets and other building blocks. In left column type a short 'description' of the text (like the name you might assign to a document, except you want to make it as short as possible (and preferably a single word).

Here is a sample, one that you can copy from here and paste into a Word document.

Signature	Roy Lasris Innovative Software Products of Virginia, LLC 117 Chisman Landing Seaford, VA 23696
Closing	It was a pleasure speaking with you today. Please let me know if you have any further questions or comments for me.
John Jones Address	John Jones 123 Main Street New York City, NY 00345
John Jones Telephone	(202) 555-1212
Mary Jenkins Address	Mary Jenkins 345 Oak Tree Lane San Francisco, CA 99976
Mary Jenkins	(567) 887-5555

Telephone	
Certificate of Mailing	I hereby certify that I mailed the attached [Type of Document] to [Opposing Counsel] on [Date of Mailing].

To create the glossary, click the Pathagoras Features drop down. Select Authoring/Editing tools and the Clause Creation Tools from the resulting screen. Click the "Table to Glossary" item.

Pathagoras will 'see' the table and begin processing automatically. You will be asked if you want Pathagoras to name each clause using the prefix/suffix style. If so, the names at the left will be preserved as the clauses' subjects. Otherwise, the name of each clause may be modified to comply with Word's bookmark naming rules. (Not to worry. The subject of the clause will be preserved in any event.)

Once the glossary has been prepared you will be asked to save it. Then follow the prompts to add the glossary to your collection of DropDown Lists and/or to a book in your current library.



Word's bookmark naming rules are fairly strict:

- The name must start with a letter of the alphabet or an underscore character.
- If you use the underscore character as the first character in a bookmark name, it will make the bookmark 'hidden.'
- The remaining characters of the name must be letters or numbers or the underscore character.
- No other symbols, including spaces or punctuation marks, are allowed.

So, if you want to retain your spaces and other special characters which can make it easier for you or your users to recognize your clause names, consider selecting the prefix/suffix naming style option. If chosen, Pathagoras will ask you for a 2, 3 or 4 letter prefix ('gen' for general is always a good one) and will assign an appropriate suffix. Pathagoras will also preserve the name you provide in the left column as the term's Subject. It is the best of both worlds. you will be able to maintain the Subject, and get a bonus: You can recall terms from the glossary using Alt-G function. Just type the prefix/suffix name of the document and press Alt-G. Instant clause.

12.6.4 Glossary from 'Painted' Document

There is not a 'right' way to create a glossary. Other sections described several techniques. Here is one that derived from a user who saw the 'apparent' structure of a glossary, but who missed the bookmarks as an essential element.

This user observed that a typical Pathagoras glossary had this structure for each term:

RedLine (*a single word representing the term name1*)

Blue Line (*one or more words representing the subject of term1*)

Body Text 1. User designed, with as much or as little formatting as desired. Body text can be as short as a single word, or long as desired. It can be multiple paragraphs; can contain bullets

and automatic paragraph numbers, headings, anything!

RedLine (*a single word representing the term name2*)

Blue Line (*one or more words representing the subject of term2*)

Body Text 2. User designed, with as much or as little formatting as desired. Body text can be as short as a single word, or long as desired. It can be multiple paragraphs; can contain bullets and automatic paragraph numbers, headings, anything!

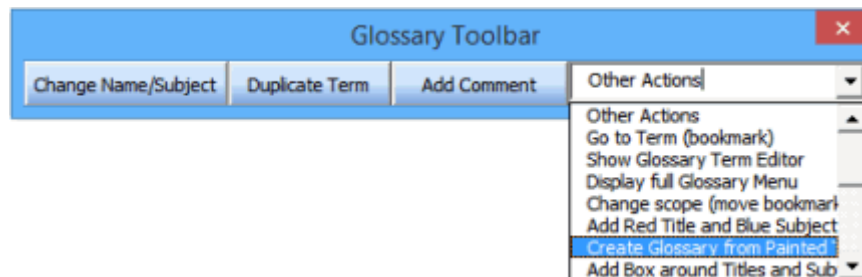
. . . . and so on.

No spaces exist between the red and blue lines or between the blue line and the first line of the 'body text'.

However, without bookmarks, the text is not (yet) a glossary.

To convert the painted document to a glossary, only a couple of steps more are needed:

1. Make sure that the glossary term names are valid bookmark names. The term must start with a letter and only letters and numbers can be used for the rest of the name. No spaces or special characters allowed, except underscores ('_') are 'legal'.
2. Save the document with the name 'Glossary' as the last word in the name (e.g., Will Clauses Glossary.docx").
3. Follow one of these two methods to add the bookmarks.
 - i. From the Pathagoras Features List, click Editing Tools | Glossary Floating Toolbar. From the Other Actions list in that Toolbar, click Create Glossary from Painted Document:



- ii. From the Pathagoras Features List, click Editing Tools | Glossary Tools. Click the Conversion tab and click the Multicolored (red and blue, of course) button that reads "Covert Painted Document to Glossary."

4. It takes only a few seconds to add the bookmarks. The bookmarks will surround the body text, and will bear the names of the red text. Your glossary is ready to go!

12.7 Clause Conversion Tools

In addition to Instant Book and Bulk Add discussed in other sections, Pathagoras offers several other powerful tools to help you create books for document assembly use. They can be found on the screen entitled "Clause Creation Tools" located under the Pathagoras dropdown menu. The tools discussed in the sub-sections here focus on the conversion of an existing book (folder or glossary) to the other kind of book.

12.7.1 Glossary to Folder

Sometimes you created (or inherited) a Glossary that you wish to return to a folder of terms. (This most often happens when you want to share your terms with an acquaintance who does not have Pathagoras, or who would be more comfortable using a folder of terms instead of a glossary.)

Pathagoras provides a simple way to convert all or a portion of a glossary into a standard folder of files:

1. Choose the "Conversion Tools" tab from the 'Clause Creation Tools' screen (Pathagoras Features|Authoring Editing|Clause Creation|Conversion Tools).
2. Click the "Glossary to Folder" option.
3. A standard looking "Libraries & Books" screen will appear, but non-glossary items will be 'greyed-out.'
4. Choose the glossary that you wish to convert.
5. A standard looking "Clause Selection" screen will appear listing all of the terms in the glossary. (The screen is slightly different in that the <Next> button is hidden, replaced by a <Glossary to Documents> button.)
6. Select the terms from the glossary that you wish to convert into individual documents and move them to the right panel.
7. When you have completed your selection, click <Glossary to Documents>.
8. You will be asked to approve the name of the folder into which the new documents will be saved.
9. Everything will proceed automatically after that. (You will see a bit of screen flashing as the new documents are created.)

12.7.2 Folder to Glossary

Perhaps a folder already contains clauses that you have been using for document assembly. Or you have a few clauses within a folder that you wish to move into a glossary because of the known advantages of working with glossaries (power, speed, editability and transportability). Pathagoras provides a simple way to add all or a portion of a folder of files into a glossary. It is very quick, and it is very simple.

1. Choose the "Conversion Tools" tab from the 'Clause Creation Tools' screen (Pathagoras Features|Authoring Editing|Clause Creation|Conversion Tools).
2. Click the "Folder to Glossary" button and follow the prompts.
3. If your clauses are already a book (as suggested in the first sentence, maybe you previously assigned a collection of documents in a folder to one of your document

assembly books), choose option #1 when presented.

If you wish to create a new book/glossary 'from scratch', choose option #2. (With option #2, you can also choose whether to convert .doc, .dot, .txt, .rtf and even .wpd files into your new glossary. However, Pathagoras can convert only one type of file at a time.)

4. Follow the prompts to name the glossary and when the checkbox screen displays, choose which of the files in the directory you wish to move into the glossary.

➔ **NOTE:** The name of the specific document/file that is currently being moved into the glossary will be proposed as the name of the glossary clause. However, all 'bookmark naming rules' (no spaces or non-alphabetic/numerical characters) are in effect during the transfer process. Therefore, if the name of the file does not meet the bookmark naming rules, you will be prompted for a proper name. (Even in this step, Pathagoras is very helpful. Pathagoras will automatically propose a better name for you.) If you prefer to pre-name all of the clauses in a folder with appropriate 'glossary names' before you begin the conversion, use the [Names & Subjects Editor](#)



➔ If you never assigned a subject, then the full name of the document will become its subject.

➔ You can import Word documents, text files, 'rtf' files.



You can even convert your WordPerfect® files into a glossary!

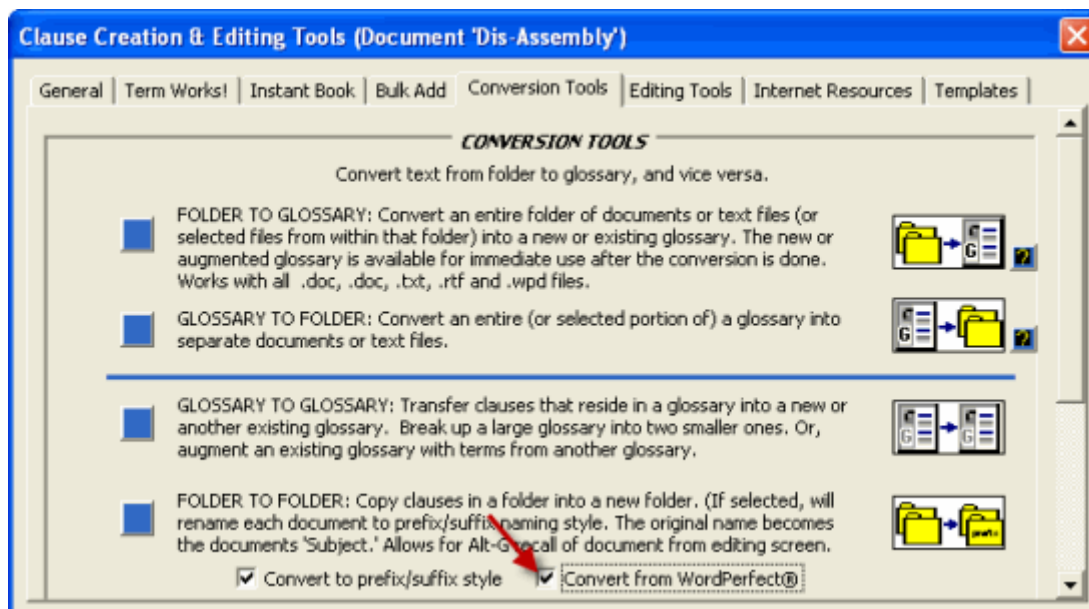
Once the above routine is complete, you should review the content of the new glossary. Edit where appropriate, rename and duplicate clauses where needed. Just be mindful of the bookmarks when editing, and stay within them. To display the bookmarks, display the Utilities/Setting screen and click the Show Bookmarks button on the top page.

➔ Any text outside of a bookmark is not included in any glossary call, so feel free to add explanatory text or helpful tips for you or anyone else who might later be editing the glossary.

12.7.3 WordPerfect to Word Folder


You can convert a folder of Word Perfect® documents to a folder of Word documents. The converted files will be written to the a sub-folder beneath the folder containing the original WP files, with identical names except for the extension (".doc" or ".docx" instead of ".wpd").

To activate this tool, click the Clause Creation Tools entry from the Pathagoras dropdown features list. Select the 'Conversion Tools' option. When the below screen appears, click the 'Conversion Tools' tab. Check the Convert from Word Perfect' box.



Note: the proper WordPerfect-to-Word converters must be installed before you can perform the conversion. These are Microsoft provided products. Normally, they are pre-installed, being a part of the standard Word installation routine. If you do not have the converter, you can download the converter here:

After you have converted the Word Perfect document to a Word document, you may need to reformat some parts of the document (for example, formatting like page breaks and columns may not convert cleanly). After you've made these formatting changes, simply save the document again (remember: it's now a Word document and will save, by default, as a Word document). There are any number of private offerings which assert that they offer more accurate conversions. If you have a large library of Word Perfect files and are making the move to Word, we certainly encourage you to try them out. Key words "Convert Word Perfect Word" is sufficient to take you to a listing of the current offerings over the Internet.

 If the converted documents do not look sufficiently like the originals, you may wish to delete the new 'doc' files and manually convert the files. Use the information and tips provided by Microsoft on their site created specifically to address these issues. <http://office.microsoft.com/en-us/word/HP052742911033.aspx>

Word Perfect® is a registered trademark of
the Corel Software Corporation

12.8 Names and Subjects

A clause can be assigned any name you wish, subject to just a few naming rules and conventions:

'Name' rules:

- **Document Naming Rules:** A document can contain any combination of letters, numbers and keyboard characters, except that the following characters cannot be used

as part of a document name: *, &, ?, /, \, :, ; You cannot use slashes or colons in your term names. If you are going to be adding terms to a glossary (as opposed to a folder of clauses), 'bookmark' naming rules will apply. This merely means that the name must begin with a letter, and contain only letters and numbers (and no spaces). The name may contain an underline ('_') character.

- **Glossary Term Naming Rules:** Glossary names have a few more restrictions. Word's bookmark naming rules apply. A bookmark name can contain only standard keyboard letters and numbers and the underscore character, and the first character must be a letter.

'Subject' rules: A 'Subject' should be provided for every clause and document so that a future operator will be able to understand the purpose and use of the term. There are no character limitations to a Subject. But you should keep the 'size' of the subject to no more than 60 characters. This is a display issue, not a rules issue.

- The 'Subject' of a document can be found under the Document Properties element. To display Document Properties, the document must be open. Click File|Document Properties. Edit freely if you wish.

➔ But see [Worldox exception](#)  604.





- In a glossary, the Subject is no more than the blue text which appears below the red 'clause name' and immediately above the actual text of the clause.

You can also add a subject to an existing document using the SaveSmart module:

- Display the document. Click SaveSmart.
- Click the Show Subject checkbox.
- Type or edit the subject in the text box provided.
- Save the document.

Editing Subjects: Because of the simple, Word based locations in which subjects are saved, it is quite easy to change them. We encourage you to do so freely. When you modify a subject, the new subject will display the very next time you display the Clause Selection Screen or print out a new terms list.

Editing names and subjects *en masse*:

- To change the names or subjects of an existing folder of terms or a glossary, use the "Names and Subjects Editor" found under Editing Tools. See [Editing Names & Subjects](#) ( [en masse](#)   428).
- If you convert a folder into a glossary, the rename occurs *en masse* as well. You can instruct Pathagoras to assign each term a new name following the prefix/suffix naming style as they are being moved into the glossary. The 'old' name becomes the 'new' subject. It is a 'best of all worlds' situation. See [Folder to Glossary](#)  318).


12.9 Styles

Unexpected style or formatting changes can have one of two distinct causes:


1. Typically they happen when the name of a style in the source document is identical to a style name in the receiving document (or template). The receiving document always controls. This 'rule' may give you unexpected results.

For example:

- Let's say that the 'Heading 1' style of the *source* document is "Arial, 18 point, bold, italics, no numbering." When you look at the source document, everything appears in order.
- Let's further say that the 'Heading 1' style of the *receiving* document (most typically, it is based on your normal.dot template) is "Times New Roman, 14 point, bold, w/ numbering."
- Since the style of the receiving document 'trumps' the style of the source document, the paragraph formatted with style 'Heading 1' will appear as "Times New Roman, 14 point, bold, w/ numbering." Your nicely laid out and unnumbered headings now appear all messed up. They are not messed up. The rule works, even when you do not want it to.

The Solution: To make sure that the receiving document always contains the same styles and layout as the source document, you should create a template based on a properly configured, but more or less 'blank,' source document. Click here for [step by step instructions](#)  98.

2. Headers and footers are controlled by a document's 'Page Setup' settings. If the settings of the source document don't match those of the receiving document, the receiving document again controls and undesired result may occur.

The Solution: To make sure that the receiving document always contains the same header and footer settings as the source document, you should create a template based on a properly configured, but more or less 'blank,' source document. Click here for [step by step instructions](#)  98.

3. Another possibility when you are experiencing style issues during document assembly when the selected book is a glossary deals with the 'scope' of the bookmark as it pertains to the source glossary term. (Ignore this discussion if the book which is giving you style issues is not a glossary. How would you know? Well, glossaries have to be affirmatively created by you or the system administrator. If you didn't create a glossary, ask the administrator. If neither of you created a glossary, then you can feel comfortable that the problem is not the one being described here.)

- When the original source text was marked out and saved into the glossary, did the marked out text include the final paragraph marker? If it did not, then the formatting contained within the final paragraph marker will not be carried into the glossary (the new 'source') or into the final document.
- By the same logic, do the bookmarks surrounding the term in the glossary itself include the final paragraph marker? If not, then the formatting that is contained within the final paragraph marker (including non-tab indents) will not be carried over into the final

document. In the latter situation (the final paragraph marker is 'outside' the bookmarks in the glossary), this can be 'corrected' by using the Glossary Editing Tools. Display the 'offending' glossary to the editing screen. Click Editing Tools|Glossary Tools from the Pathagoras dropdown features menu. Select the 'Tools and Utilities' tab and click the <Paragraph Markers> button. Follow the prompts. Pathagoras examine each term to determine if the ending bookmark encloses the paragraph marker of the last substantive paragraph.

Word provides two very helpful keyboard shortcuts which may help in your quest of making the documents and styles uniform.

- **Alt-Space:** As you have noticed, characters can be formatted independently of the paragraph within which they reside. (That way you can make letters within a word a *different* **color** or **font**s. You can reset a character (or group of characters) within a word to the style of the underlying paragraph by placing the cursor within the word and pressing Alt-Spacebar. (Or highlight a group of words. All highlighted words will be reset to underlying the paragraph style.
- **Alt-Q:** Word makes it easy to add non-font changes to a base style. Example: indenting, alignments, spacing. You can 'clean' a paragraph of such changes, and cause it to revert to the base style by pressing Alt-Q. So, if the base normal style is "Times New Roman, 12 pitch, no indent" and the current paragraph is "Normal + Indent .25 Before", pressing Alt-Q within the paragraph will remove the "Indent" and restore the paragraph the the standard "Normal" configuration.
- **Clear Styles:** If you activate the Styles menu and click the "Clear Formatting" selection (near the top), all style definitions of the selected text will be cleared, and the text will be assigned to 'Normal' style. This is sometimes the easiest way to start from scratch.
- **Copy styles:** You can easily copy styles to and from your Normal.dot to another template or active document.
 - In Word 2003 and prior versions, click the Tools menu and then click 'Templates and Add-Ins'. Click the 'Organizer' button toward the bottom of the screen and then click the Styles tab. Copy items to or from a different template or file, click Close File to close the activedocument and its attached template or to close the Then click Open File, and open the template or file you want. Click the items you want to copy in either list, and then click Copy.
 - In Word 2007/2010, click the 'southeast' pointing arrow in the Styles box (Home tab) to bring up the Styles menu. Click the Manage Styles icon at the bottom of the screen. (It's the third one from the left. It's not labeled, but you can hover over the icon to reveal its purpose.) Click the Import/Export button at the lower left. Click the items you want to copy in either list, and then click Copy.

This takes some (but very little) practice to get the feel for how all of this works, but these tools are very helpful as you purge your base documents of what often times are complex combinations of styles and fonts that serve no purpose.

See Also

Templates Q & A

[Assign Template to Book](#) 

[Assign Template to Document](#)  102

Styles (FAQs)

[Paragraph Styles \(Templates\)](#)  103

12.10 Automatic Paragraph Numbering

12.10.1 General Information


A common feature of any word processing system is Automatic Paragraph Numbering. The feature works in the following helpful ways:

- If a new paragraph is added to a document, it is given the proper next sequential number.
- If a new paragraph is inserted between currently numbered paragraphs, the paragraphs beneath the insertion are properly renumbered.
- If a paragraph is deleted, the paragraphs beneath the deletion are properly renumbered.

There are practically an infinite number of styles that you can insert into your document. The basic ones are plain list numbering (1., 2., 3., etc.), outlining (I., a., i., II., a., b., etc.), and 'legal style' (1.1, 1.2, 1.2.1, etc.).

Pathagoras works with any version and any style of Word's automatic paragraph numbering systems. It works with any off-the-shelf system designed to tame Word's sometimes challenging systems.

Regardless of which system you decide to use, keep the following in mind.:

- If you are assembling documents from separate documents (a folder of forms), make sure that the switch for the first paragraph in each form that allows numbering to 'Continue Previous List' is turned 'on'. Otherwise, you may end up with all paragraphs numbered "1."
- All of the 'first' numbers in each separate document will *appear* a "1". This is regardless of whether the Continue Previous List is on or off. It is when the separate documents are pulled together as one that the need for Continue Previous List become obvious. Do not forcibly set numbers in second-, third-, fifth- etc. -in-order document to the ones you think they should be. This will just mess things up.
- If you use one of the 'Instant Book' or 'Bulk Add' tools to create a folder of documents, and assuming that the master document is properly numbered, then you should be in good shape. The 'Continue Previous List' settings in the source document will carry over automatically into each document created from the master.
- Make sure that the numbering schemes for the target document are consistent with the schemes for the original/source document. If the numbering schemes are not identical, the number of the newly inserted text will restart with '1'. See the [next section](#)  325 for the techniques to repair this.

Here are two excellent references for 'beyond simple Automatic Paragraph Numbering':

1. <http://www.shanakelly.com/word/numbering/numbering20072010.html>
2. <http://blogs.office.com/b/microsoft-word/archive/tags/bullets+and+numbering/>

12.10.2 Re-numbering Issues

As a 'paragraph assembler' at heart, Pathagoras handles any manner of automatic paragraph numbering schemes. [See previous section](#)³²⁴.

However, Words imposes a requirement that automatic paragraph numbers must follow the same 'scheme' within a document in order for the automatic renumbering magic to occur.

Paragraph numbers which are of different schemes will not 'blend.' The separate schemes retain their separate identities, and a set of numbers from one scheme will display numbers independent from other sets in the same document.

"But my automatic numbering styles look identical! Why don't they renumber properly." or "When I bring in a new paragraph or section of text, the new numbers start at 1?"

While the numbering schemes may appear identical, if you are having this problem, it almost certainly is because the schemes are not identical. Word (unfortunately) will detect what is even a subtle difference.

One way to insure a common numbering scheme is to insure that all of the assembled clauses come from the same source. If you disassemble a document into its individual components (either manually or via [Pathagoras' 'document dis-assembly' tools](#)³³⁰), you are assured of 'same source/same scheme.' When you reassemble the clauses in any order, the automatic paragraph numbering will be proper.


Another way to insure a common numbering scheme is to make sure that your 'normal' template contains the numbering scheme that you desire, that all other users who might share clauses for the document assembly system use the identical scheme. Further, you and others must be diligent to not vary from the scheme. That means no adjusting those sliders at the top to change margins, not adding tabs, etc.

Keep in mind: this 'care' only applies when you intend to assemble documents from individual clauses created by various users AND where you wish to preserve automatic numbering for those inserted clauses. If you only build documents from complete templates, you needn't worry about this consistency of schemes.

If you see that the numbering is restarting at "1" in the middle of your assembled document, you have to presume that the documents have different numbering schemes. To harmonize the schemes, try one of these two techniques:

(1) Assemble (or otherwise gather together) in a large document all of the paragraphs that need to be harmonized. Make sure that a single numbering scheme is applied to all paragraphs. (If all paragraphs are to be numbered, just highlight the entire document, click numbering and select the proper scheme. If less than all of the paragraphs are to be numbered, use the paint tool (it's the 'paint brush' in the toolbar) to copy the numbering scheme from paragraph to paragraph. **Tip:** Double clicking the paint tool will keep the formatting in active memory. Each click applies the 'paint'.)

(2) Open up one document (we'll call this document #1) that you know has the desired scheme. Open up a second document (we'll call this #2) that does not have a compatible scheme. Cut the text from #2 and paste into #1 at the end. Make the numbering scheme compatible (see the paragraph immediately above for instructions and a tip on how to do so). Then cut the new text out of document 1 and paste back to #2. Save #2. Repeat for all other documents. (We are working on a tool to accomplish all of this automatically, but it is a challenging thing to program.)

 **Note:** When you disassemble a document into its component parts (paragraph (1) above) or try the one-document-at-a-time technique of (2) above, the first numbered paragraph in each separate document that you save out will necessarily begin with '1'. Don't let this scare you. When you assemble a document using one of those clause, the proper number will be applied.

13 A Different Approach

The previous and subsequent topics suggest using options, multiple choices and groupnames to help you to make the various selections your ultimate documents will require.

There is an alternative approach that many using Pathagoras have adopted. We are not recommending this as a preferred approach, but as you study this more, you will see inherent benefits, including:

Not having to program much in the way of options and optional text, and pronoun 'switchers'. Since they are preassigned as part of the Client Intake process, there is no need to have anything other than plain, simple variables.

Only variables that are in the specific document under construction are used during the personalization phase of document assembly. This is a more intuitive approach to many.

Only the variables within the document (and that appear in the IDB screen are processed. (In the other method, where all variables are displayed, Pathagoras will process each variable in the list to see if it resided in the document. While the process of determining whether a particular variable is in the document is very quick, there is still some delay when hundreds of variables are being processed. Here, since far fewer variables are actually processed, there is a time savings.

Drawbacks: Pathagoras cannot write back to the source data record. Therefore, if changes to the data are expected, you should not use this method to display the data.

Here is how it works.

Create a 2 column table (a simple Word table will work)

In the left column, type all the variables you think that you want to collect, including 'administrative' values (Client Name, Client address) which you want to manage clients in general, and more document specific variable names (Testator Name, testator sex, child@1 name, etc.) that might used in your documents. Don't worry at the beginning if you don't capture all variables. You can easily add more later. Pathagoras can handle 999 variables.

Here is what the start of the simple table might look like.

BTW, you can use Excel to create a similar table. Just make it two columns. Pathagoras will be able to import it.

To import the table, activate the Instant Database (Alt-D). Click Power Tools and click the Import Button.

Choose the appropriate button to import the data. Pathagoras will automatically add brackets to the variable names. Save the record with an appropriate client/customer name.

When you are creating documents, use the variable names that appear in left column. Here you will want to use the square brackets to denote the variables.

Save you document in the appropriate file locations. (Pathagoras never dictates this. Store them where you want.)

When you are ready to assemble a document, display the document on the editing screen. We of course recommend that you use one of Pathagoras document assembly tools (clause selection screen or DropDown List or Alt-G. That way, Pathagoras will call up a copy of the document, not the original. If you call up the original, be sure to SaveAs the document as soon as possible so you don't forget (and accidentally overwrite the original).

When you are ready to replace variables, press Alt-D to bring up the Instant Database screen. Normally we would have you call up the client's record at this point, but this is the 'different approach.' Click the Scan button to display just the variables in the on-screen document. Then, look for the small checkbox just to the left of the Data records list in the upper right corner of the IDB screen. Click it. Now select a data record. Very quickly, Pathagoras will pair any variable showing at the left with it value contained in the data record. **ONLY THE DOCUMENT VARIABLES WILL BE DISPLAYED.**

The Pathagoras System

Document 'Dis'-assembly

Part




XIV

14 Document 'Dis'-assembly

Document 'dis-assembly' is the process by which an existing document is deconstructed into its component parts. The individual pieces are saved as separate clauses ('building blocks') in a designated folder or glossary. These pieces form the clauses from which a wide variety of new documents can be assembled.

Building documents from 'building blocks' (clauses) is one of the two basic approaches to 'document assembly.' The alternative method is 'template based' assembly. There, you start with a complete (actually, overbuilt) document and, by answering questions strategically placed within the document, you remove those portions of the template that are not needed for the particular project.

Which approach you use is one of the more important choices you will make in setting up your system. Each has its advantages. Click [here](#)  for a full discussion and comparison of clause-based ('building blocks') versus template based document assembly.

You probably already have many documents that are ideal for dis-assembly. The ones found in your 'office forms' folder certainly qualify. So do the ones found in treatises and form books to which you subscribe, or those that you received at Continuing Professional Education courses.

Pathagoras provides several document dis-assembly methods ranging from 'manual' (simple highlight a section of text, copy and paste it into a new document and save that document into your book to highly automated (place simple markers such as "(*)" at the document 'break-up' points and press a button to disassemble to document into the predetermined sections.

The 'easiest' and most basic (but still powerful) disassembly techniques are discussed in the following pages. All techniques for disassembling documents into building blocks are discussed and fully illustrated in the separate pamphlet called [Document Disassembly \(Creating Building Blocks from Whole Documents\)](#).

Click

14.1 Manually

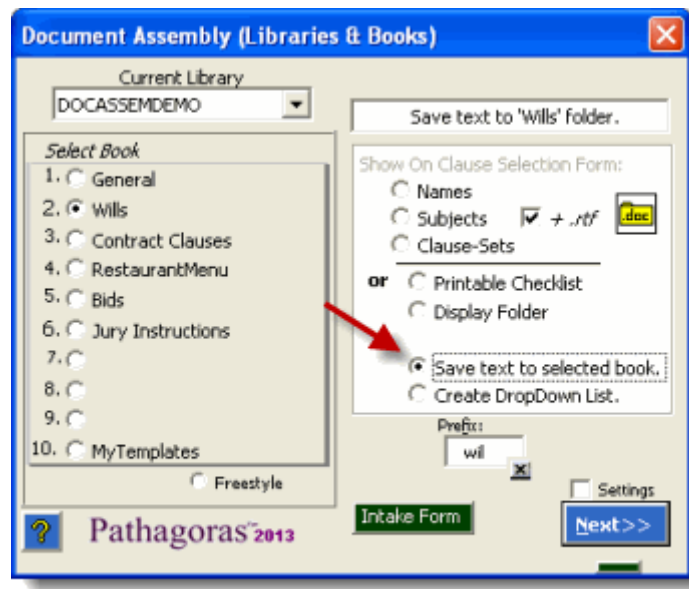
Document disassembly can be nothing more than highlighting a piece of text that you want to save as a separate clause, pasting it into a new document, and then saving that new document into the appropriate folder. The next time you display a book using the Document Assembly button, the new document will instantly and automatically display alongside the other documents in the folder. (If a DropDown List is using that folder as the source of its content, you may have to 'Refresh' the DropDown List for the new document to show. But this is a one-time only refresh.)

The disassembly techniques discussed below are simply automated versions of the above process.

14.2 Using Libraries & Books Screen

Highlight the text you want to save. Click the Document Assembly button. When Libraries & Books Screen displays, and select the book into which you wish to save the highlighted text. The screen expands to show you several choices. Click the one titled "Save text to selected

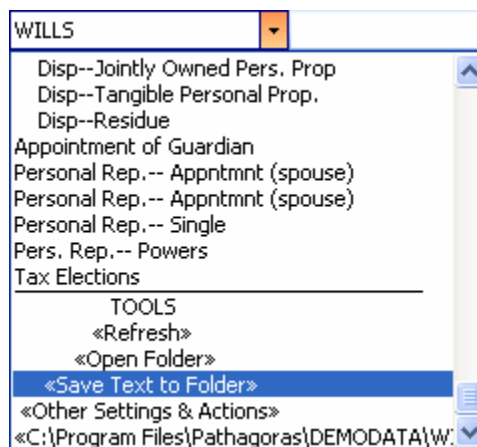
book.'



Press Next and follow the prompts to name the document and provide an optional subject.

14.3 Using DropDown List

Highlight the text you want to save. Click on the DropDown List into which you wish to save the highlighted text. Locate the element in the Tools section near the bottom of the list titled 'Save text to Folder'.



Follow the prompts to name the document and provide an optional subject. The new document will be present in the DropDown List then next time you display it as well as being present in the standard Word/Windows Explorer screens if you later choose to navigate that way.

14.4 Bulk Document Dis-assembly

The above techniques work fine when the items you want to 'dis-assemble' are few. But if you want to disassemble an entire 20 page contract or trust document into perhaps 50 (or 150) component parts, another approach is advisable.

'Document Dis-assembly in Bulk' is an alternative to the one-at-a-time approach discussed in the previous pages. It requires a good bit of document preparation (to mark out the begin and end points of each clause-to-be, to assign a title and a subject to the new clauses. But once the prep is complete, Pathagoras can perform the actual disassembly work for you automatically.

Bulk Document Dis-assembly is discussed in a separate pamphlet which can be downloaded [from this link](#).

The Pathagoras System

Term Works!

Part



XV

15 Term Works!

Term Works! has already been touched on in earlier section of this Manual in the context of ['Document Disassembly'](#)³³⁰. But Term Works! offers an incredibly simple 'highlight & add' feature to make it easy to quickly move text into a book of your choice.

Term Works! is activated whenever you press <Alt-G> against highlighted text. (You can also activate Term Works! from the Pathagoras features menu.)

Your choice of "Highlight & Add" versus a more deliberate Document Disassembly approach depends upon the particular project you have in mind. If you are adding a single piece of text, or just a few pieces, to a book, then Highlight & Add is definitely the way to go. Nothing could be quicker or easier. If, however, you need to add multiple building blocks of text from a single document, definitely consider the document disassembly procedures.

15.1 Screen Tour

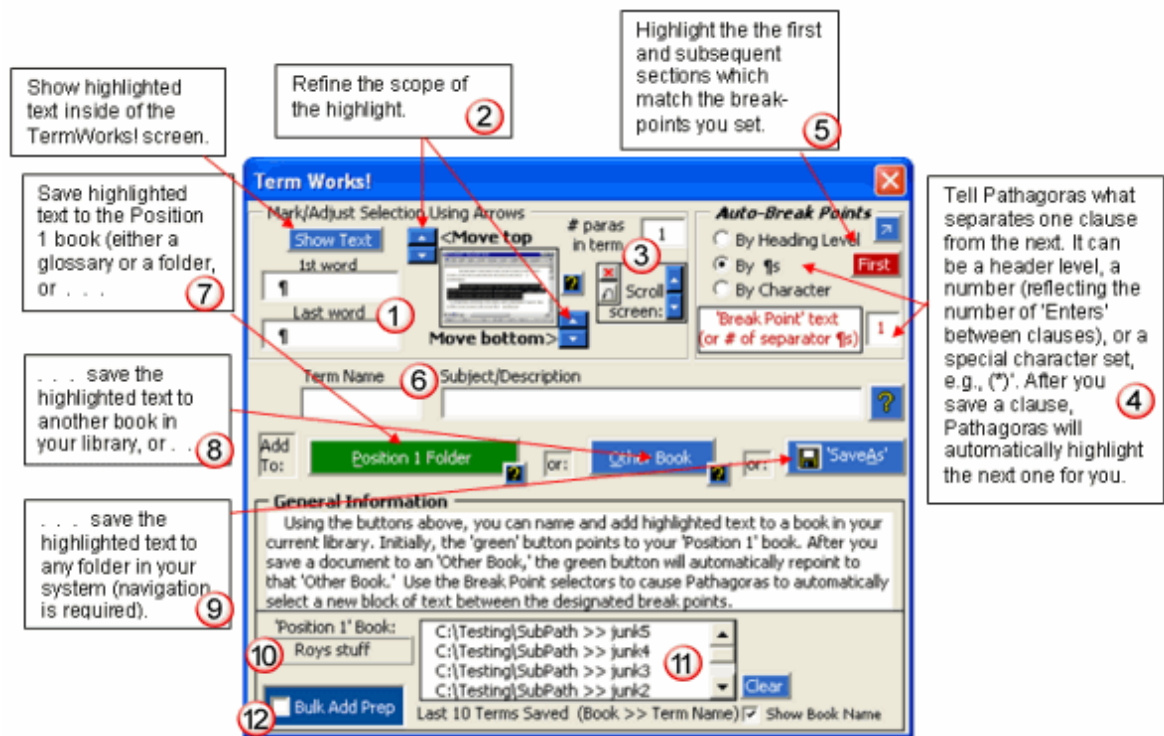


Figure 1. The initial Term Works! screen.

Let's first take a brief tour of the major screen elements. Although a bit busy, it provides a substantial amount of information and options in a small area. All screen elements are functional and useful.

The upper part of the **Term Works!** screen reflects, sets, or allows you to manipulate, the boundaries of the text you wish to save.

(1) shows the first and the last words of the text.

(2) allows you to adjust the scope of the highlighted section of text. (Note the two up/down arrow sets, one for the start point of the highlight, the other for the end point.

The <More> button, when clicked exposed 'left' and 'right' adjusting arrows to even

further refine the selection.)

(3) Allow you to scroll the editing screen up and down without leaving the *Term Works!* screen

(4) allows you to set 'Break Point' criteria to enable Pathagoras to automatically identify successive sections of text for manipulation. (The "Break Point" field is discussed below.)

(5) Allows you to highlight the First and then subsequent sections of text that meets the criteria set in (4) below.

The middle section:

(6) provides space for naming the term and giving it a subject;

(7), (8) & (9) allows you to select the 'target' book into which you want to create a new term based on the highlighted text.

Typically, the 'green' button will read 'Position 1 Book,' reflecting the book occupying position one of the current library. If the term you are saving is named following the 'prefix/suffix' style, and if the prefix is registered, 'Position 1 Book' (button 7) will be replaced with the book associated with the prefix. See Figure 7 below, and the text surrounding Figure 7.

The <Other Book> button (#8) takes you to another screen where you can direct Pathagoras to save the text in (a) one or more books from the current library, (b) the SuperGlossary (if assigned), (c) the SuperFolder (if assigned) or (c) the same folder in which the underlying document has been saved.

(Note that you can also click "Save As" (9) and move text into a folder 'free-style' and without book restrictions.);

The lower section of the screen is primarily informational.

(10) Advises you of the name of the book that occupies 'position 1' in your current ('default') library. (The 'Position 1' book is reflected in the 'green' button in most cases. See below for the exception. This makes adding to the 'Position 1' book that much faster.

(11) lists the last 10 clauses that you saved. (Click on any name in the list and the selected item is copied into the Name box. Edit as appropriate.)

(12) resets the screen to allow you to take advantage of TermWorks! automated Bulk Add markup tools discussed and illustrated in the separate [Document Disassembly](#) pamphlet.

15.2 Selecting/Refining Scope

Once the Term Works! screen is displayed, you first should decide how Pathagoras will determine the initial section of the text to highlight. (This assumes that nothing is currently highlighted.)

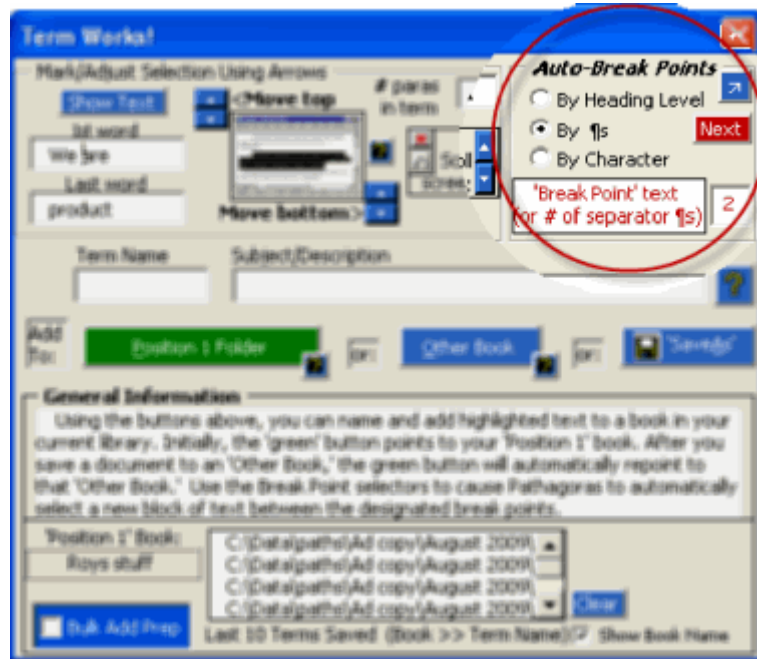
Your choices are:

(1) by document header;

(2) by the # of paragraph markers ('enters') between sections; or

(3) by a special character set. For this last choice to work for you, you must pre-typed into the source document those character at each location that you want a break to occur.

In the below example 'By Paragraphs' has been selected. The number of paragraphs which Pathagoras will look for to determine a section break is 2 (i.e., a 'double space'), and that value has been inserted into the box in the lower right quadrant of the Auto-Break section of the screen.



If the Next button is 'red,' click it. ('Red' just means that no text has yet been selected.) Pathagoras will highlight the first section of text in your document that meets the desired criteria. In the example we have chose, this will be the first block of text that ends with 'double-spaces.'

If you want to further refine the selection, use the up/down arrows in the upper left side of the screen. See below figure. This lets you expand and contract the scope of the highlight so that just the text you want is marked. (In a typical document dis-assembly session, Pathagoras will automatically highlight the next block of text after the last one was saved, but you may still wish to refine the scope of the selected text.)

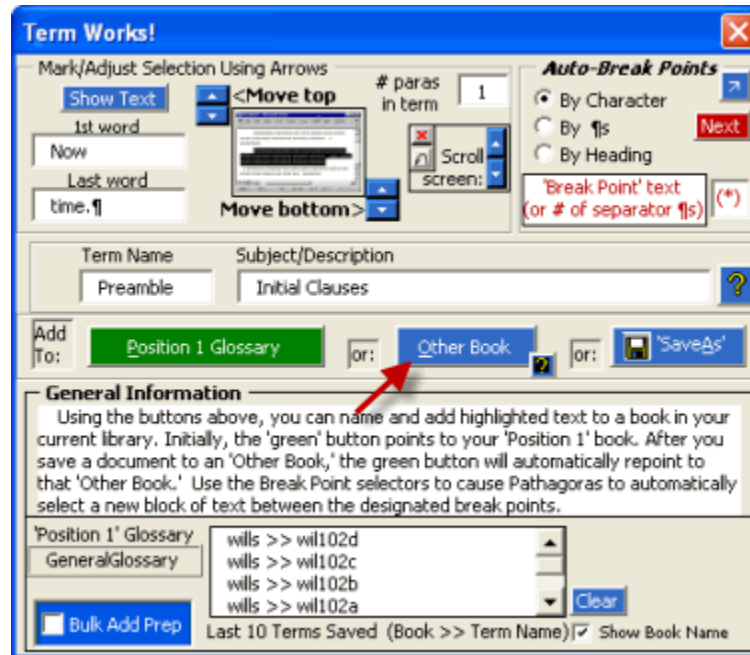


Figure 2. Highlight the text. Then provide a name and a subject for the new term. If Position 1 is not the desired location, click <Other Book> to view available target books into which new clause can be added.

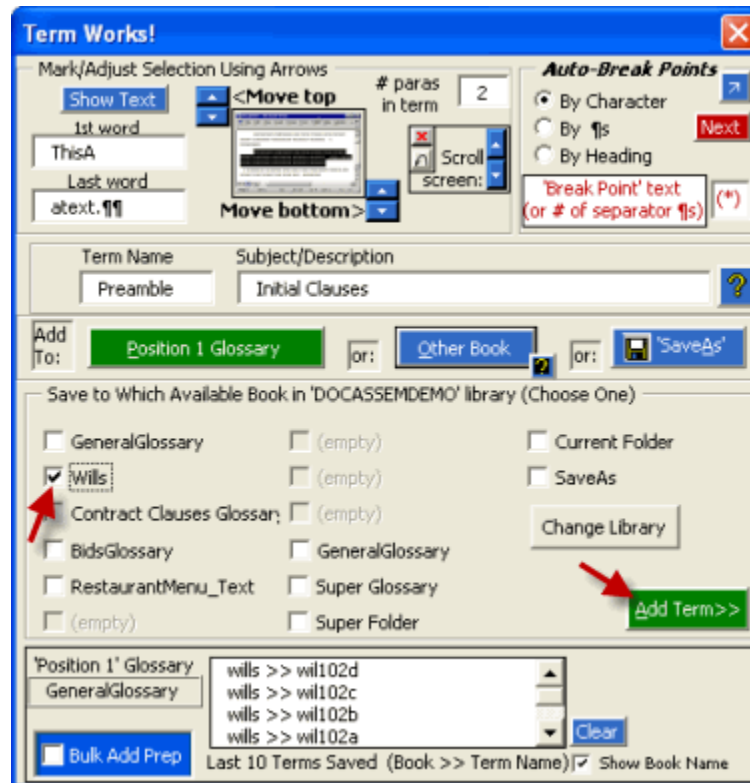



Figure 3. Check the specific target book(s) into which you wish to save the new clause. Click Add Term.

Using this and other tools, you can further augment and refine the book to create a powerful, yet easy to modify and easy to understand, document assembly system.

Term Works! also can help you add clauses in rapid fire succession. All you need to do is provide either (1) the number of blank lines ("Enters") between each clause you want to add, or (2) provide a special character set that you have used to mark up a document for 'disassembly'. With this information, after you add one clause, Pathagoras will quickly jump to the next one, allowing you to quickly add a series (even a long series) of clauses into a book. Click [here for a more detailed, illustrated discussion](#)  334.

15.4 Creating New Book

Creating a new book with Term Works!

You will typically use **Term Works!** to add clauses to *existing* books. However, in the very early stages of your system's development, you may be creating new books from scratch.

1. Display the document from which you want to draw your first clause.
2. Highlight the text that you want to be the first clause. Example follows:

This is 'pretend' highlighted text. This text can come from any source--an existing document, copied and pasted into Word from the Internet, anywhere.

For this exercise, we want to add this text as the first term in the book. To do so, we

(1) highlight it (with the mouse, drag the cursor from the top left to the bottom right of the desired text. Word will 'paint it' the reverse colors to indicate the highlight) and

(2) Press the keyboard combination <Alt-G> 'against' the highlighted text. The Term Works! screen (Figure 1) will quickly appear.

Term Works!

Mark/Adjust Text

1st word: This

Last word: appear.¶

paragraphs in term: 1

Buttons: <Move top, More>>, Move bottom>

Term Name:

Description/Subject:

Add To: Position 1 Glossary or: Other Book BulkAdd Prep.

General Information

Using the buttons above, you can (1) add highlighted text to a book in your current library (2) instantly add highlighted text directly to your 'Position 1' book, or (3) mark-up a document for a later 'BulkAdd' of the terms. To create a one-touch button to access this screen, click the green button below. Press a <?> button for more information.

'Position 1' Book: GeneralGlossary

Create Term Works! ToolBar. [Green Button]

Close

Highlight text before calling up this screen. Use arrows at left to adjust/re-mark. Then press button below representing the desired function or destination of the highlighted text.

Figure 1. A 'blank' Term Works! screen.

Note that the first and last words of the highlighted text appear near the top of the form.

The next steps are to (1) name the clause, (2) provide a subject to more fully identify the purpose of the clause and (3) select where to save it. Because we are adding a new book, the middle "Other" is an appropriate choice.

Term Works!

Mark/Adjust Text

1st word: My first term

Last word: Initial steps in 'book creation' process

paragraphs in term: 1

Buttons: <Move top, More>>, Move bottom>

Term Name:

Description/Subject:

Add To: Position 1 Glossary or: Other Book BulkAdd Prep.

General Information

Using the buttons above, you can (1) add highlighted text to a book in your current library (2) instantly add highlighted text directly to your 'Position 1' book, or (3) mark-up a document for a later 'BulkAdd' of the terms. To create a one-touch button to access this screen, click the green button below. Press a <?> button for more information.

'Position 1' Glossary: GeneralGlossary

Create Term Works! ToolBar. [Green Button]

Close

Highlight text before calling up this screen. Use arrows at left to adjust/re-mark. Then press button below representing the desired function or destination of the highlighted text.

Figure 2. Provide name (1) and subject (2) and then click 'Other' (3).

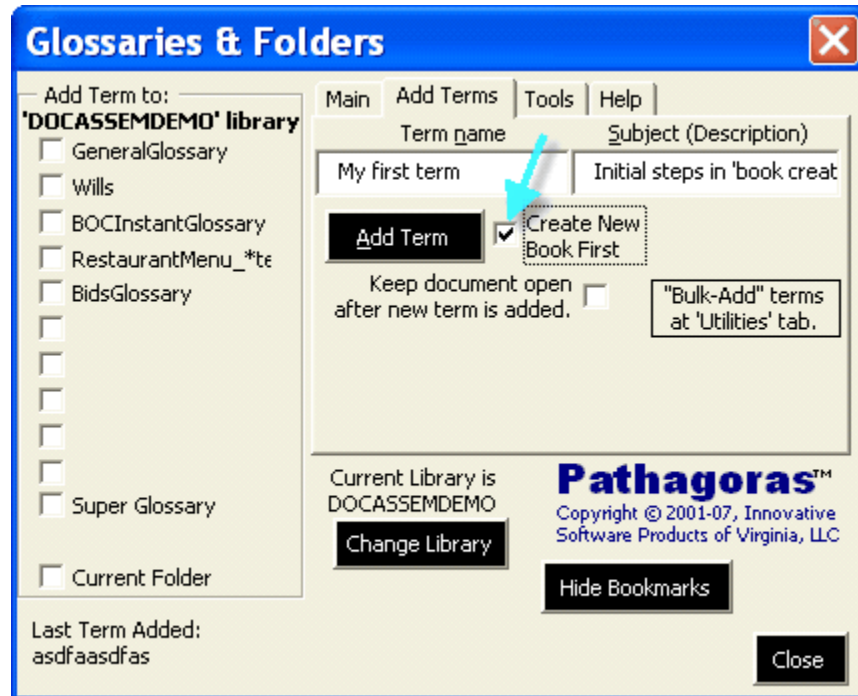


Figure 3. Since we are starting a new book, click the “Create New Book First” box. Then click Add Term.

Pathagoras will ask you whether you want the book to be a glossary or a folder of terms. See the discussion above as to which is ‘better.’ Then it will ask where you want to save the new book. Navigate to the desired location. Pathagoras will save the text in the new book, using the name and subject you provided.

15.5 Prefix/Suffix Naming

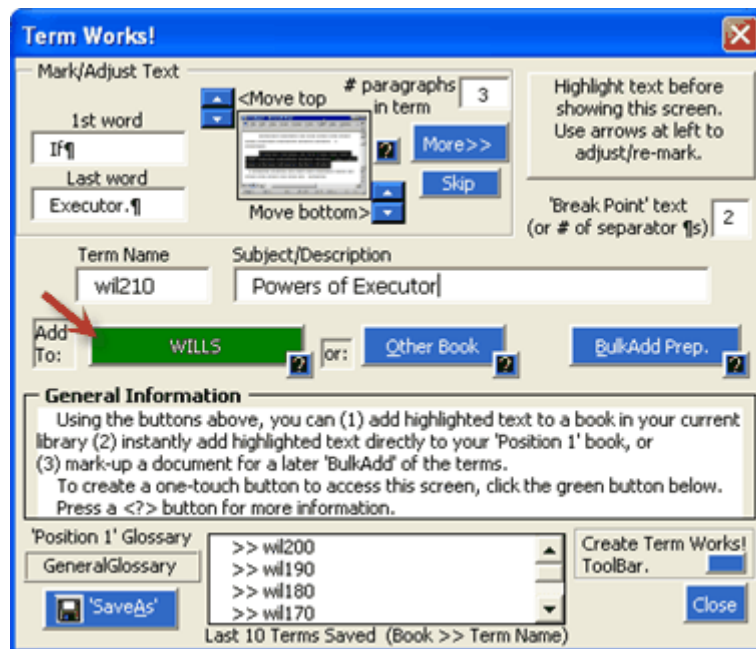
Pathagoras, Term Works! and the Prefix/Suffix Naming Convention.

- We strongly recommend that you use of the Prefix/Suffix naming style for naming your documents. The main reasons are set out at this link: [Prefix/Suffix Naming Convention](#) ⁴²⁰
- And more to the point in the lesson covered here: If prefixes are currently in use, and if you want to add a new clause to your book named with the appropriate prefix, Pathagoras' Term Works! screen will automatically point you to the desired book.

Let's say you have text that you want to add to a book which follow the prefix/suffix style. (We must also assume that the prefix has been registered. This is an easy step, but you are referred to the section of the Manual referenced above for those steps.)

Highlighted text you want to add. Active the Term Works! screen. Type the name of the clause using the prefix/suffix style. Pathagoras will automatically 'sense' the name as being in the prefix/suffix style and will re-point the green button to the proper book.

Here is a picture. The prefix for the WillGlossary is ‘wil’.



When the term name with prefix 'wil. . .' is typed, Pathagoras locates the WillGlossary and offers to save the new term there.

15.6 Bulk Add connections.

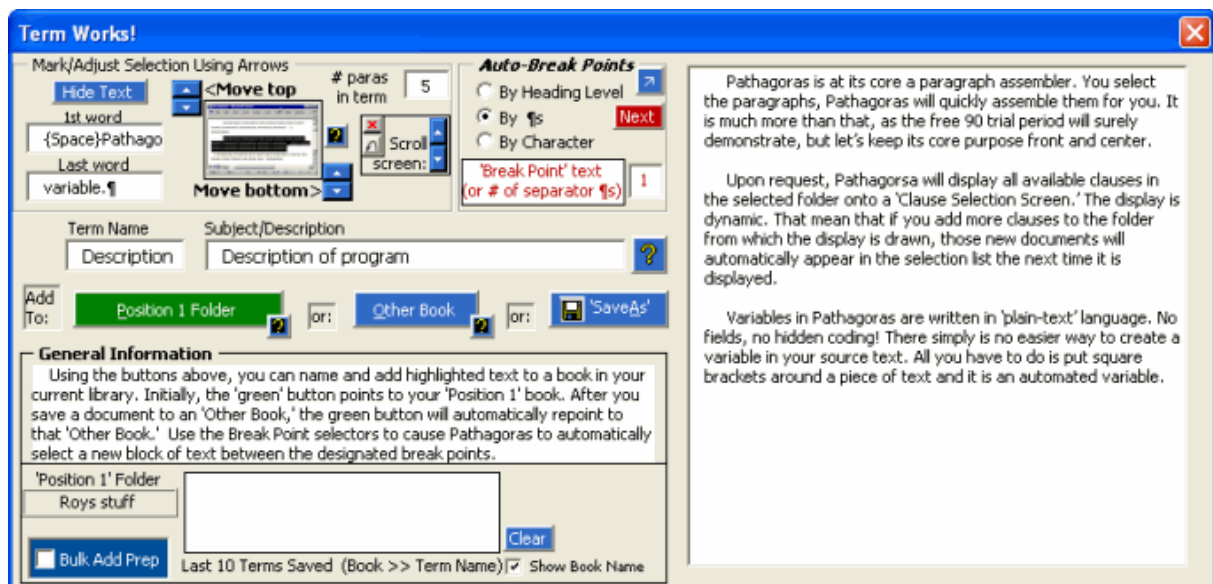
Term Works! and Bulk Add.

You can use **Term Works!** to help you to pre-mark clauses in preparation for using the Bulk Add procedure. So, instead of manually typing the 'begin' and 'end' markers into the master document, you can use **Term Works!** This technique is more fully discussed and illustrated in the separate pamphlet [Document Disassembly](#).

15.7 Show Text



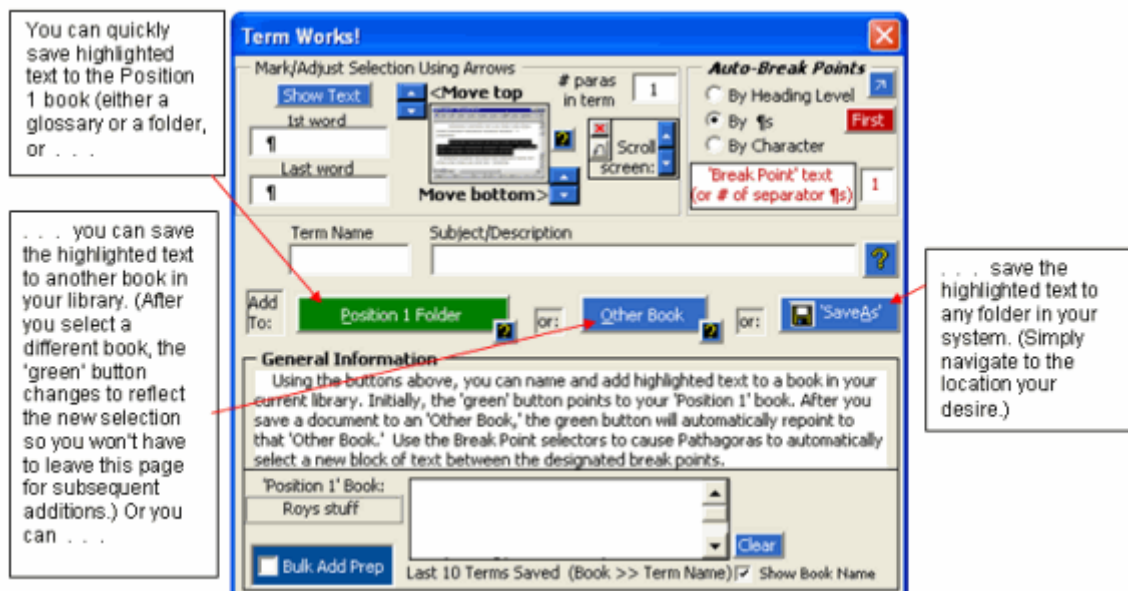
If the Term Works! screen covers up too much of the underlying text (or you simply want to see the text that has been selected), click the <Show Text> button. The TermWorks! screen expands and displays the highlighted text in a box to the right. See image below.



Whenever you change the scope of the highlighting, the text shown at the right will also change. (Note: You cannot edit text in this box. You can only do that within the actual document.)

15.8 Add text Where?

Once you have selected the first block of text that you want to break out, you need to decide where you are going to save it. The Term Works! screen offers 3 choices:



(1) The 'Position #1' book. This refers to the glossary or folder that sits on bookshelf #1 in your active Library. The actual name of the glossary or folder is shown at the lower left quadrant of the screen, next to the 'Last 10 Terms Saved' listing;

(2) Another book in your current Library; or

(3) Any where else you please. If you are a beginner and have not created any books just yet, you should just click the 'SaveAs' button.

When you click any of the above buttons, Pathagoras will transfer the selected text (with all formatting retained) to a new document. Then, if you selected:

(1) (Position #1 book) Pathagoras will insert the text into a new document in the selected folder (or a new term in the selected glossary.)

(2) (Another book), Pathagoras will open a screen displaying your current library, from which screen you can select the appropriate book into which to save the text.

(3) (SaveAs) Pathagoras will display a standard Word 'SaveAs' screen and allow you to navigate to the location to which you wish to save the text. (Pathagoras requires no special location for its clauses, so the choice is yours. The default is going to be the folder in which the source document resides. Our suggestion: create a sub-folder here and save the text in that sub-folder.)

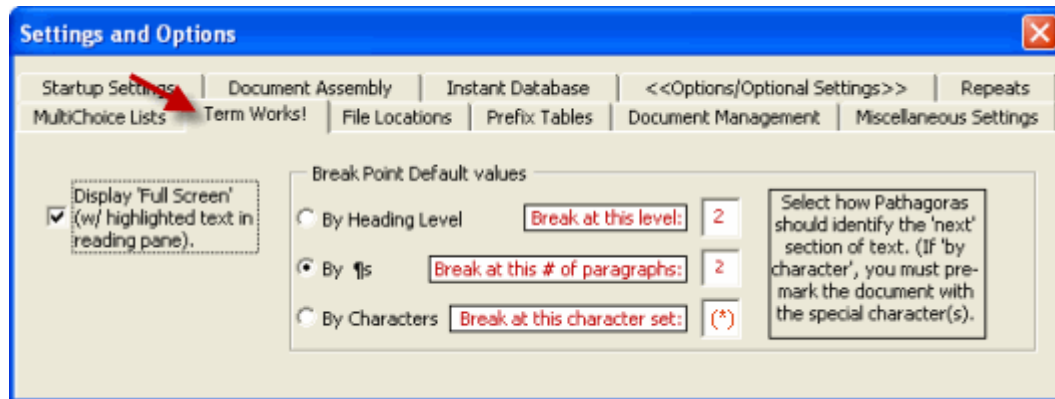
Once you have completed the save, Pathagoras will offer a 'Continue process?' option. If you select it, Pathagoras will save and close the new document, return to your 'source' document, and highlight the next block of text that comports with the set parameters. This is the 'automatic' part, and makes your dis-assembly of a document incredibly easy.

Note one other thing: The long green button beneath the Term Name box now reflects the name of the book or of the SaveAs folder into which you saved the initial clause. If you wish, you can just click on the green button to save your next clause(s).

15.9 Setting Term Works! Defaults

You can set the default settings and appearance of the Term Works! screen via Pathagoras comprehensive 'Settings' screen:

1. Call up the Utilities/Settings screen.
2. Click the <All Settings> button.
3. When the Setting and Options screen appears, click the Term Works! tab. This is what you will see:



Check, click and complete the desired 'default' values (the values that the Term Works! screen will 'open' with the next time you call it up) and close the screen.

15.10 Hints and Tips for Document Disassembly

Do I have to complete the Subject field?

Well, it's not *mandatory* that you do so, but we recommend that you always complete the subject field. It can never hurt. Plus, as you are building more and more sophisticated systems, you will find that having pre-planted a subject, even in your early documents, will be very useful to you.

Maybe this will help you to decide. Think of the purpose to which you are going to put the various pieces of your dis-assembled document. If you don't plan to recall terms using the Alt-G hot key, and plan only to recall terms via a DropDown List or Clause Selection screen, you can get away with not attaching a subject. (When you don't assign a subject, Pathagoras will still do it for you, attaching the documents name, sans the .doc or .docx extension, as its subject.)

Let's assume the following. The document you want to disassemble is a list of names and addresses. (Remember, Pathagoras can handle anything!) Here is a sample list.

John Q. Adamson
Suite 232
12345 Main Street
Jamestown, VA 23696

George P. Washburn
443 Apple Tree Lane
Mt. Vernon, VA 23876

Thomas W. Jeffers
543 Independence Blvd.
Monticello, VA 23567

What names do you want to assign to each address? Of course you could call the first one "Adamson, J.Q. Address" and you will know exactly what it is, but that is a lot of typing for a supposed 'shortcut.' Even better: 'jqadamson', or 'adamsonjq' or even better still: 'jqadd' (the latter reflects the initials plus the suffix 'add' to indicate an 'address'. (Following this pattern, you could use 'jqaph' to represent a phone number.)

It is simple to type any of the the above to the screen and press the hot-key combination <Alt-G>. Instant address!

But how will you or others in your office recall all of your addresses. Some you will memorize. Others you will need a list. And that's where the Subject line comes in. Pathagoras can maintain the relationship between the 'short' name and the 'complete' subject. So, while the document or glossary term name might be jqadd, you can display the subject "Adamson, JQ." in any DropDown List or Clause Selection screen. It really is the best of both worlds.

Note:

The '(*)' method of marking up text to show major section breaks is used in several modules within Pathagoras. It is used within TermWorks! (discussed in this section, within the Bulk Add routine (next) and within the Instant Book routine (second next).

Keep in mind that '(*)' (or whatever character set you choose) is just text. It is definitely not 'code'. It has no special meaning to Pathagoras beyond marking a begin and end point where these functions might look to perform their respective functions.

The Pathagoras System

GotForms?

Part



XVI

16 GotForms?

GotForms? (<Alt-S>)

Have you *GotForms?* You probably do. *GotForms?* is so titled because the answer is typically 'Yes' even before you began using Pathagoras. If you have any forms on your computer where the variables are denoted by [words between brackets], or by underscore characters, then you have *GotForms!* And even if you don't have such forms, they are (obviously) very easy to create.

GotForms? and Instant Database work in very similar ways. *GotForms?* is a "one-term-at-a-time" replacement tool. Instant Database is an "all-at-one-time replacement" device. IDB uses the same scan engine used by *GotForms?*, but places what it finds onto the left hand column of the Instant Database screen. Instant Database can identify, present, search for and replace up to 900 separate variables in a single sweep.

Exception: Because you need 'context' to determine what is needed to complete a variable denoted by underscores ("_____") you must use *GotForms?* to fill in blank lines.

See Also:

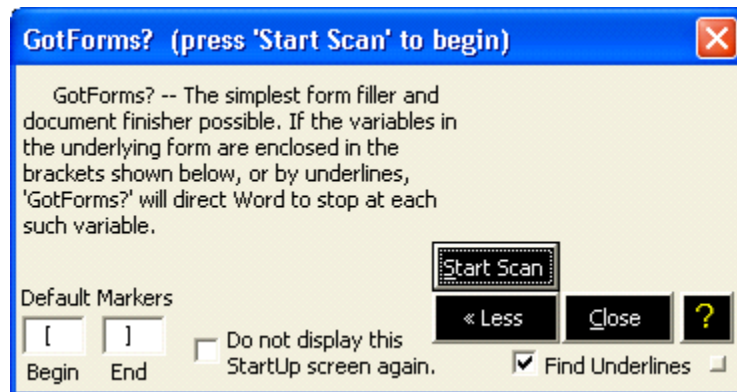
[Instant Database](#) 

16.1 Standard Screen

Completing variables with *GotForms?*

(The following exercise presumes that you are displaying a new or existing document on the editing screen in which either bracketed variables or underline-variables exist.)

1. Press <Alt-S> (for 'scan') from your keyboard. This screen will appear:



GotForms? (initial display after <Alt-S> is pressed)

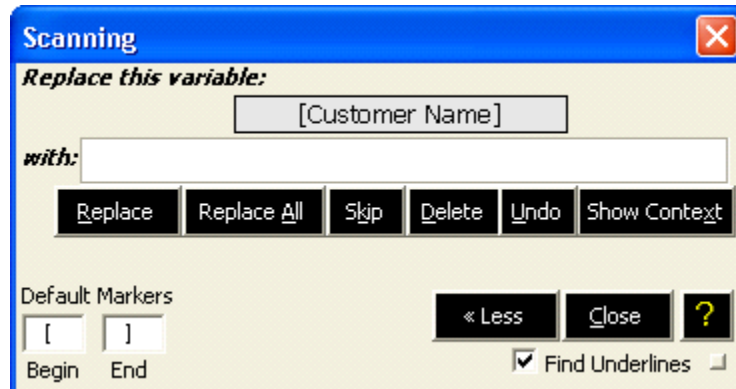
Note: You are not limited to searching only for square brackets. You can choose any 'bracket' set you want. Type any begin and end markers you wish to use into the boxes at the lower left side of the screen.



You might consider marking some of your source documents so that a *GotForms?* scan will pick up one set of variables and Instant Database scan

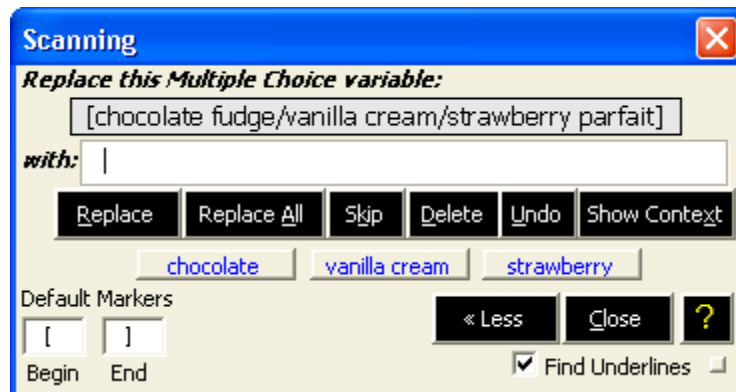
another set. The more you become familiar with the strengths of the two systems, the more power you will see they bring to your word processing environment.

2. Press the <Start Scan> button near the center of the screen. *GotForms?* will scan the document for brackets variables or underlines and stop at each to give you the opportunity to substitute personal values in place of the variable. In the below example, *GotForms?* 'found' the variable [Customer Name]. All you need to do is type the appropriate replacement value in the text box in the middle of the screen. Then, click <Replace> to replace just the current variable or <Replace All> to replace each occurrence of the variable in the document. *GotForms?* will automatically move on to the next variable.



GotForms? showing a simple variable

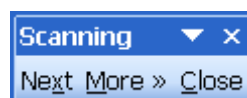
3. If a multiple choice variable is encountered, the various choices (sometimes truncated) are presented as selectable buttons on the *GotForms?* screen:



GotForms? has encountered a multiple choice variable in the text. Note selection buttons.

16.2 Minimized Screen

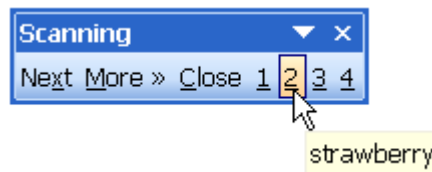
If you would prefer a smaller screen from which to work, click the "<<Less" button on the *GotForms?* screen. (It sits to the left of the <Close> button.) Here is what you will see:



Here are the differences:.

- The main *GotForms?* screen disappears and the smaller screen (above) replaces it.
- Your keyboard's Tab key turns into a variable-to-variable navigator:
 - Press tab (or the Next button in the screen) and *GotForms?* proceeds to the next variable.
 - Type the replacement text.
 - Press tab, replace the next field, etc.
- This is similar to Word's built-in text navigator, but the differences are marked. First of all, Word requires you to protect the document to take advantage of field-to-field navigation. Pathagoras does not. Secondly, if the variable you encounter is a multiple choice variable, the multiple choices will appear as buttons on your screen. Try that with plain Word!

If a multiple choice variable is encountered while minimized, the variables are presented as numbers just to the right of the "Close" button. You can determine the content of each choice by hovering the cursor over the number:



16.3 Show Context

Because an variable represented by a series of underscore characters (e.g., "_____") has no inherent meaning, when you encounter one, you may wish to press the "Show Context" button on the *GotForms?* screen. The underline will be displayed between two asterisks, and you will be able to see where the underline appears in relation to the surrounding text.



Figure 1. The GotForms screen has identified an underscore variable. Click the 'Show Context' button to display the surrounding text (Figure 2).

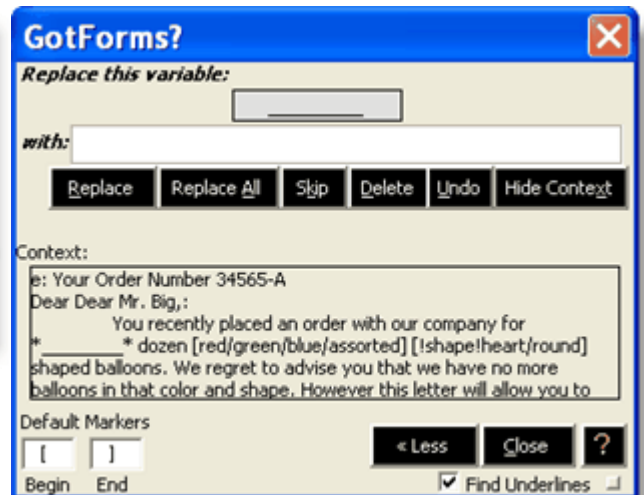


Figure 2. Now you can tell that a quantity has been requested.

16.4 Underlines as Variables

The *GotForm?* module can not only process variables it finds between brackets (e.g, [Client Name]), but it can also process underlines as variables. Of course, not all underlines are intended to be variables. For example, a signature line (like that shown below) of let's say 50 underlines in length would not typically be a variable. The date block next to a signature line is likely not a variable either. But the line to the left of Seller probably would be.

...

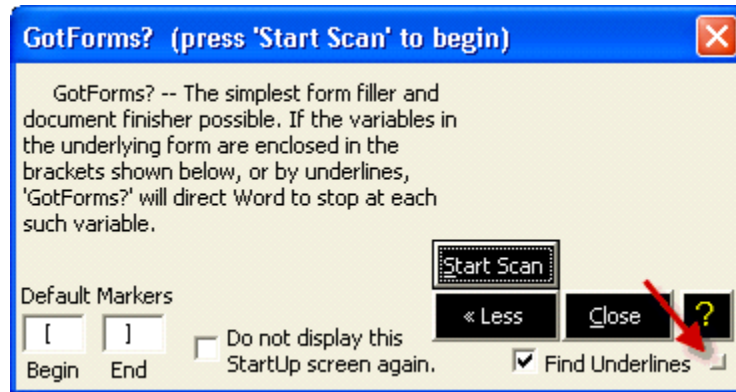
Witness the following signatures:

_____/_____/_____(Date)

_____, Seller

Pathagoras provides a way to exclude from processing those underlines which fall outside of a designated range. All underlines *within* the range are processed as variables. Those *outside* of that range are ignored.

To set the minimum and maximum values, click the small box to the right of the 'Find Underlines' checkbox.



When you click the small box, Pathagoras will proceed to ask you for the minimum and maximum number of underline characters which you want to constitute a 'variable.'

16.5 GotForms?: When To Use?

If Instant Database is so good, why should I ever use *GotForms?*

Our experience is that most customers never use and likely never will use *GotForms?*. The Instant Database is just too powerful and flexible to require the use of this auxiliary tool in most situations.

However, there are a few circumstances where *GotForms?* will be a life saver. Consider these:

- Variables as underlines: this is discussed in the [previous section](#)³⁵¹, so we invite you there.
- You only have a couple of variables in the document and just want to quickly replace them and you don't need or want to save out the variables and their replacement values as an Instant Database record.
- You need the context to complete the variable:

By using sufficiently long groupnames and/or 'colon phrases' to introduce variables, you can almost always give the end user a sufficient contextual basis for making a selection or completing a variable. But sometimes the process can get out of hand. Consider the following text:

The (husband or wife) shall be entitled to use and claim all of the payments made to the IRS in (Tax Year) which were sent as estimated quarterly tax payments for (Quarterly Tax Pmts For), including payments made which reflected both parties social security numbers and {wife or husband) shall sign all appropriate releases to (husband or wife] to enable (him or her) to claim the appropriate credits.

Neutering this can be quite a challenge. If you try to give meaningful group names to each variable, you might end up with something like this:

The [!spouse entitled to prior tax payments!husband/wife] shall be entitled to use and claim all of the payments made to the IRS in [year for which spouse is entitled to tax payments:Tax Year] which were sent as estimated quarterly tax payments for [Year Associated with Quarterly Tax Pmts], including payments made which reflected both parties social security numbers, and [!spouse forgoing interest in prior tax payments!husband/wife] shall sign all appropriate releases to [!spouse entitled to prior tax payments!husband/wife] to enable [!spouse entitled to prior tax payments!him/her] to claim the appropriate credits.

This setup will work. Try it: copy the above and paste it into a document and run the Instant Database against it.

Nevertheless, this may be too complex a setup. When these variables appear in the Instant Database screen, they are much longer than the space typically allotted. Plus, it is just a lot of text. If you feel that a simpler setup is better, consider this:

The [**husband/wife**] shall be entitled to use and claim all of the payments made to the IRS in [**Tax Year**] which were sent as estimated quarterly tax payments for [**Tax Year**], including payments made which reflected both parties social security numbers, and [**husband/wife**] shall sign all appropriate releases to [**husband/wife**] to enable [**him/her**] to claim the appropriate credits.

Something like the above is 'perfect' for **GotForms?** and requires practically nothing in terms of setup. When the document is scanned by **GotForms?**, the user can see all of the variables in context, making it quite easy to complete, and that is what GotForms? allows.

Note the following: The variables are introduced by a regular bracket followed by two asterisks. In another section of the program -- in the Instant Database setup screens -- a 'double asterisk' is the default 'ignore' this text for an Instant Database scan. Therefore, the variables in the above example will not be displayed in the Instant Database screen after you press <Alt-D>. These are viewed as 'final cleanup' type variables and are typically looked upon as the 'exceptions'.

The Pathagoras System

DropDown Lists

Part

XVII

17 DropDown Lists

One of Pathagoras' most powerful document assembly tools is also one of its simplest, both in setup and in use. It is the **DropDown List**.

A generic "drop down list" is any of those lists that reside at the top of your Word editing screen that give you quick access to Word settings. For example, Word presents various 'styles' and 'fonts' in drop down lists. Simply point and click to a new style or font and the texture of the document changes.

Pathagoras' "**DropDown Lists**" work in much the same way, but instead of changing the look of the document, you use the elements of the List to insert selected blocks of text into your document under construction.

Here is a summary of what these Lists are and what they can do.

- A Pathagoras DropDown List is a standard drop down list that resides at the top of your editing screen.
- DropDown List allow you to retrieve documents, text snippets, images and other items with a simple 'point and click' action.
- The target folders of DropDown Lists are standard Windows folders.
- You can display up to 10 DropDown Lists at a time. (The currently visible Lists are referred to as a 'Collection'.)
- You can maintain up four separate collections of DropDown Lists. Therefore, up to 40 DropDown Lists can be simultaneously maintained. You can easily switch among your Collections via the Collections panel.
- Once assigned, no navigation is required to retrieve documents (or other files) reflected in a DropDown List..
- The target folder can be *any* folder. The target folder can contain any file.
 - In most cases, the folder will contain Word documents (either entire documents or building block type files).
 - But folders can also contain text files, images (.jpg or .gif, .tiff, etc. files), Excel® spreadsheets, PDF files, WordPerfect® documents or . . . , well, you get the picture -- anything.
- A DropDown List can also reference the contents of a glossary (a single document containing dozens or hundreds of separate terms). The same 'non-restrictions' apply to glossary DropDown Lists as apply to those representing folders.
- You can activate a 'Tree Service' feature, allowing the display of the contents of the parent folder and, in two clicks, the contents of any child folder beneath.
- Creating each DropDown List takes 30 seconds tops. Once created, the List remains always active, always visible and always ready. When you exit Word and then return, so do the DropDown Lists.

[Create DropDown List from Document Assembly Screen](#)  357

[Create DropDown List from Clause Selection Screen](#)  357

[Create DropDown List 'free hand'](#) 358

[Creating and using DropDown List 'Collections'](#) 369

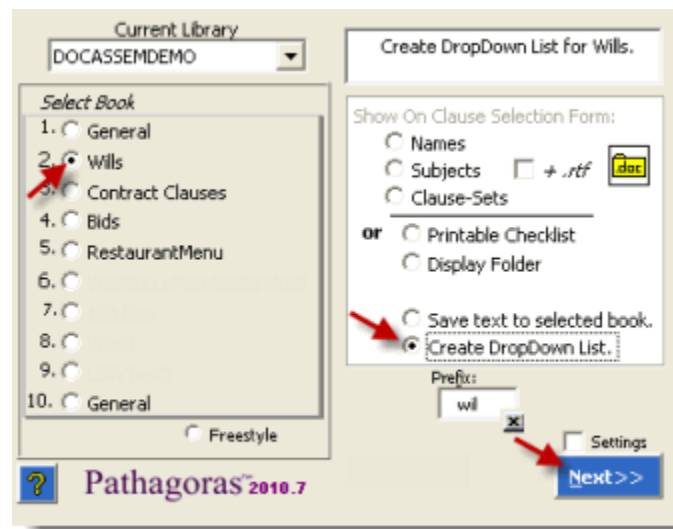
17.1 From Document Assembly

To create a DropDown List from an existing book,


Display the Libraries & Books screen and select the book from which you wish to create a DropDown List.

The action box opens in the middle of the Libraries & Books screen. Click “Create DropDown List” option.

Click **Next>>**.



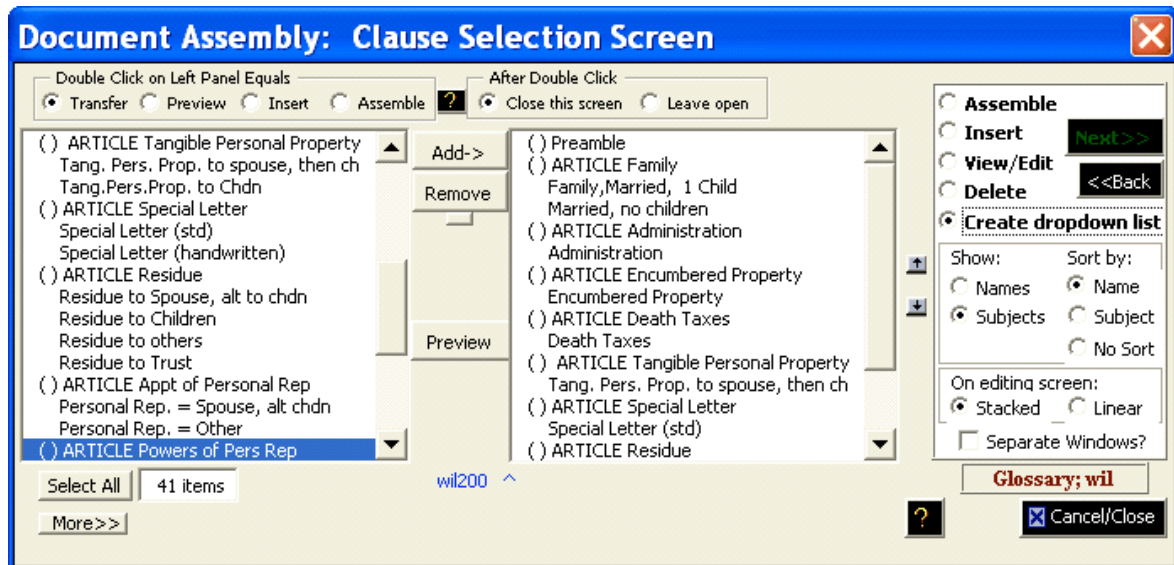
Create DropDown List from a book in a library.

 If you want the DropDown List to comprise some, but not all, items in a particular book, create the List from the Clause Selection Screen. That is discussed in the section that immediately follows.

17.2 From the Clause Selection Screen

You can create a DropDown List from the Clause Selection screen. A radio button for just that purpose is at the far right side of the screen.

➔ Creating a DropDown List from the Document Assembly screen (previous section) places *all* items in the book into a drop down list. This method allows you to select *specific items* to be inserted into the list.

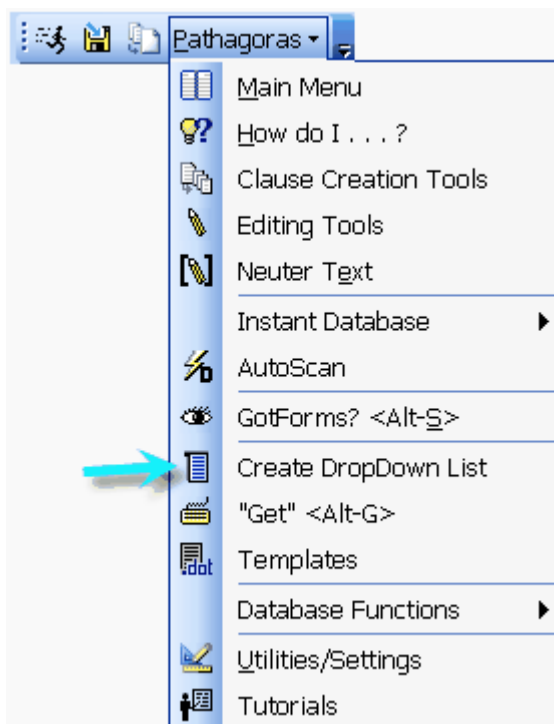


Create a DropDown List from a tailored selection of elements.

17.3 'Free Hand'

Creating a DropDown List 'freestyle:'

Drop down the Pathagoras features menu and click the "Create DropDown List" entry.



Follow the prompts. Select the type of content you want the DropDown List to contain. Your choices are:

documents, templates, spreadsheets, images, Word Perfect files, PDF files, and others.

Navigate to the folder that contains the files that you wish to be inserted into the

DropDown List. (By folder, we are referring to any standard Windows folder that already exists on your computer or across your network.)

If you choose a folder of Word documents, you will be asked whether you want the list to display the list using 'Names' or 'Subjects.' If you have assigned subjects to the documents in the folder and want to use them for the display, select 'Subjects.' In almost every other case, select 'Name.'

You will be asked to give the list a title. The default title will be the name of the folder which contains the terms. You can accept that, or type in anything else. Make sure that the list name is descriptive not only for you but for others who may need it.

That is all.

See Also:

[Using DropDown Lists](#)  362

[Image Assembly](#)  378

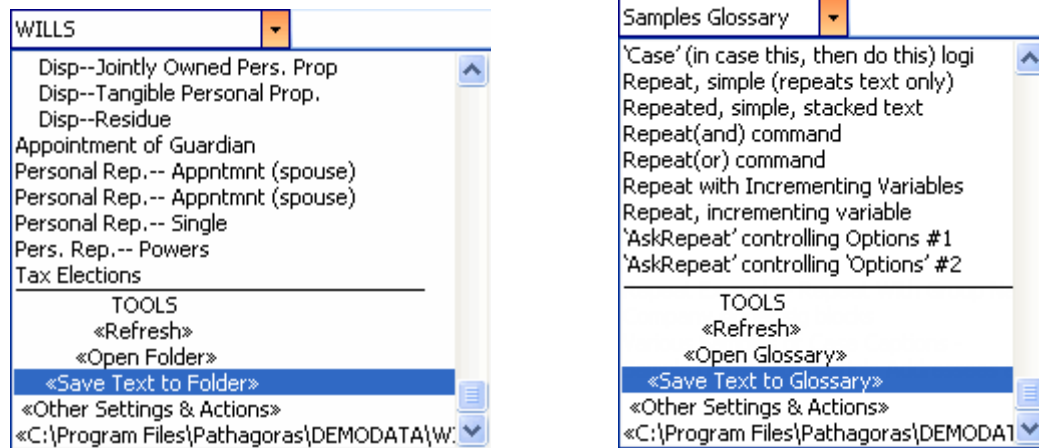
[PDF Assembly](#)  378

17.4 Adding Content to DropDown Lists

Pathagoras provides a very simple method to add new text into the Drop Down List. (Actually you are not adding text to the List so much as you are adding content to the source folder or glossary that the list represents. See notes below illustrations.)

You can add clauses to a folder or glossary represented by an existing DropDown List using the list itself. Here are the steps.

- If the entire document on your editing screen is what you want to add, simply click the «Save Text to Folder» (if the DropDown List is a folder) or «Save Text to Glossary» (if it is a glossary). See figures below.
- If you want only a piece of the active document to be added to the List, highlight with your cursor the portion you want and then click «Save Text to Folder» or «Save Text to Glossary», as appropriate.
- Pathagoras does the rest. Provide a name and a subject and you are done!
- Don't worry about 'Folder' vs. 'Glossary.' Pathagoras knows which is which, and will only display the proper option based on the source of the List.
- There is absolutely no navigation required on your part, and you don't have to «Refresh» the List after you make an entry.




Adding text to folder (left) or to a glossary (right) can be done with Point & Click simplicity.



To further illustrate the elegance of this feature, we propose the following exercise:

1. Create a DropDown List for a folder to which you intend to add text (we'll call this the 'incomplete folder').
2. Display a document that contains some text that you want to add to the 'incomplete folder.'
3. Highlight the text you want to move into the folder. Click the «Save Doc to Folder» entry in the DropDown List.
4. Repeat.

 We emphasize that when you 'add content' to a DropDown List, you are actually adding the content *to the folder or glossary* that the Drop Down List represents. The new content is placed into the source folder or glossary, and it is there for all document assembly purposes. Therefore, if you look at the source folder or glossary (maybe by clicking «Open Folder» or «Open Glossary») you will see your addition there. If you perform a 'standard' document assembly routine, the new term(s) will appear in the Clause Selection Screen.

Note: As an alternative to the above, you can directly add documents to the folder (or new terms to the glossary) using any of the other methods that Pathagoras or Word make available.

[Click here](#) ³⁰⁶ for a comprehensive list of all 'add content' tools. However, if you use one of the alternative methods, you should click «Refresh» to update the content in the DropDown List.

17.5 Repointing a DropDown List

If you move the contents of the folder to which a DropDown List currently points, the List will (of course) become dysfunctional. You will need to repoint the list so that it can 'grab' the files from the new location.

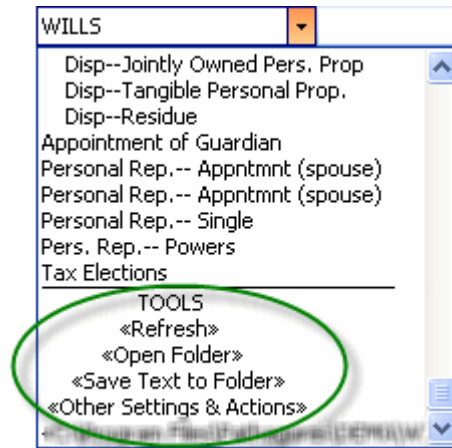
A couple of options exist.

Repoint via the Libraries & Books screen:

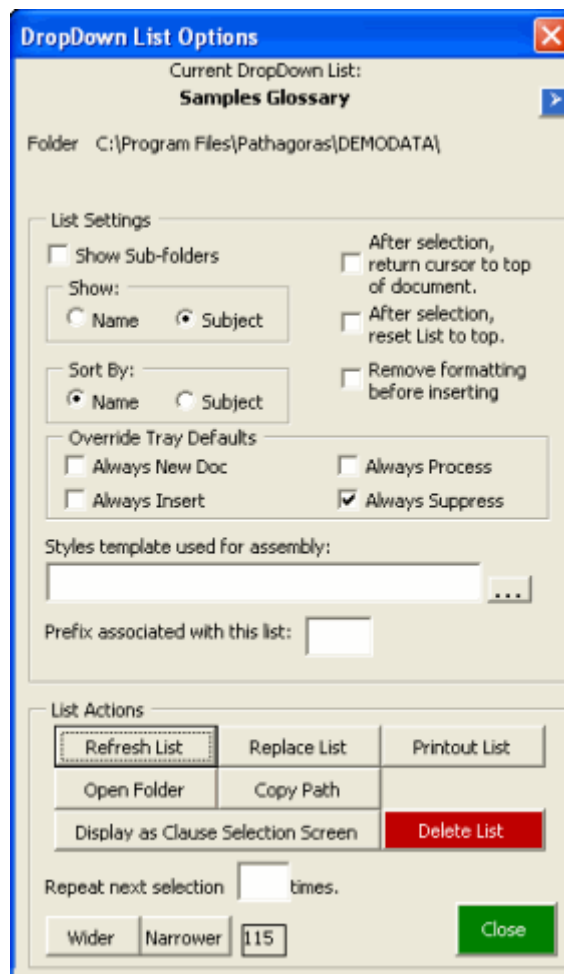
If the list reflects the contents of an existing book, you can call up the Document Assembly 'Libraries & Books' screen (the one that appears when you click the Document Assembly button. Click once on the book name. Then click the "Create DropDown List" option. Pathagoras will instantly create the List and ask which position you want the List to occupy. Select the appropriate 'tray' and 'position' of the current list. That is all.

Repoint via the actual DropDown List:

As a practical matter, the repointing function is actually a 'replacement' of the new pointer for the old. Drop down the DropDown List. Scroll down to and click **«Other Settings & Actions»** at the very bottom of the List.



This is the result"

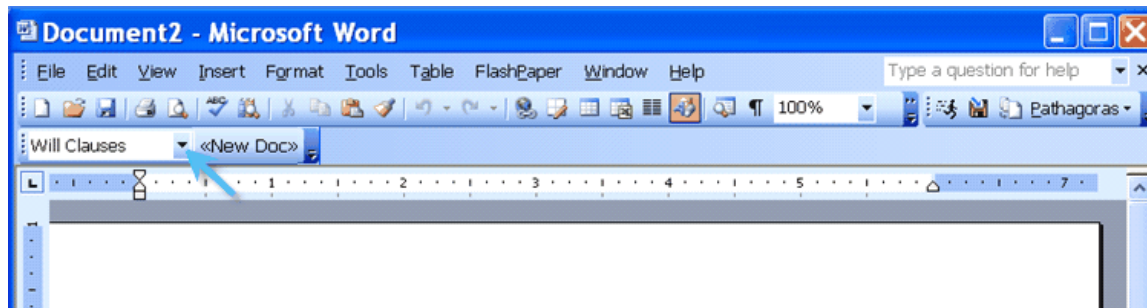


Click the 'Replace List' button. Follow the steps to navigate to inside the folder at the new location. Select one file from the folder's content (it does not matter which one). Click 'OK' to lock in the folder name and you are done.

17.6 Using a List

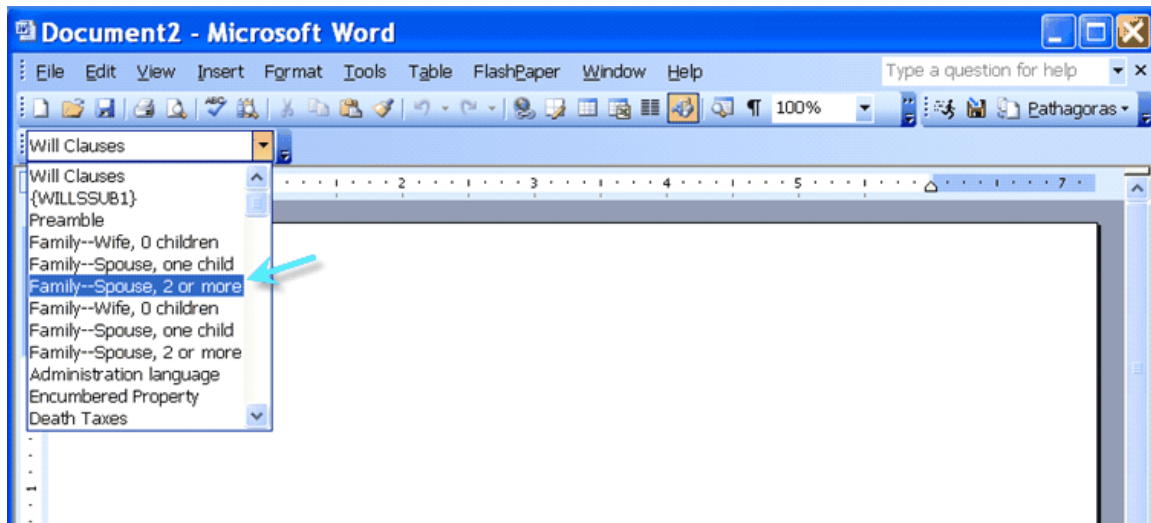
Using DropDown Lists

Regardless of which method you used to create the DropDown List (methods discussed in earlier sections), Pathagoras will transfer the names of all of the documents in the source into a DropDown List. It is now on your screen (look in the menu area). Once it is on screen, it is ready to use. Just point and click.



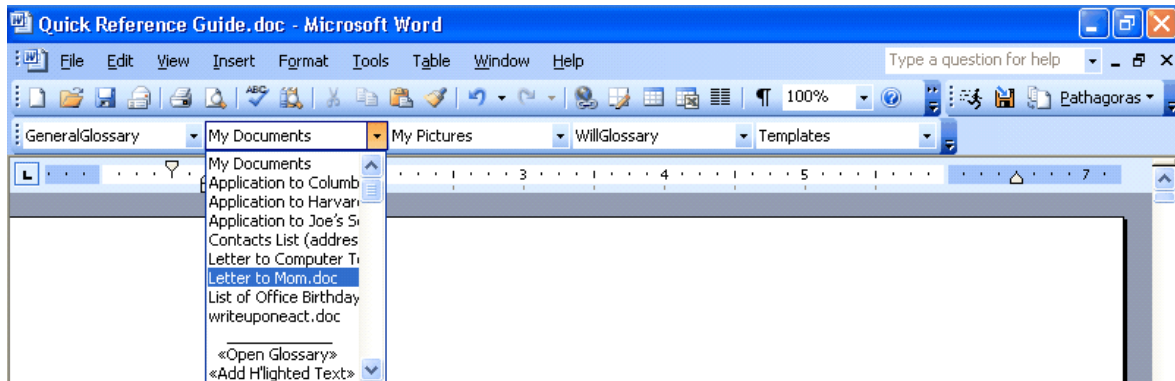
The DropDown lists displays in the Menu area.

Point and click to insert any item contained in any DropDown List, depending upon the state of the «Insert/NewDoc» toggle at the left side of the DropDown List Controls Panel.



DropDown list 'dropped down'. Select a clause.

Up to 10 Lists can be simultaneously presented. That should make it quite easy to assemble complex documents from a wide variety of sources. Drop down a list, select an item, drop down the same list or another list, select an item, and so on. If you mix images and charts with regular text, you will find Pathagoras to be an incredibly powerful tool to use.



OTHER COMMENTS:

- A DropDown List ("DDL") is a document assembly tool. It presumes that you are building or adding to a new document. Only a copy (and never the original) of the selected item is inserted into the active document. But all formatting from the original is maintained.
- Because it is so easy to create, delete and recreate DropDown Lists, don't get 'hung up' on how 'best' to create one. Just do it! Play with them until you get the hang of it all.
- Pathagoras doesn't care whether a DropDown List contains links to complete documents, to individual clauses, to pictures, spreadsheets, PDF files, or whatever. If Word has the capacity to insert a particular kind of file, it can be fed into a DropDown list.

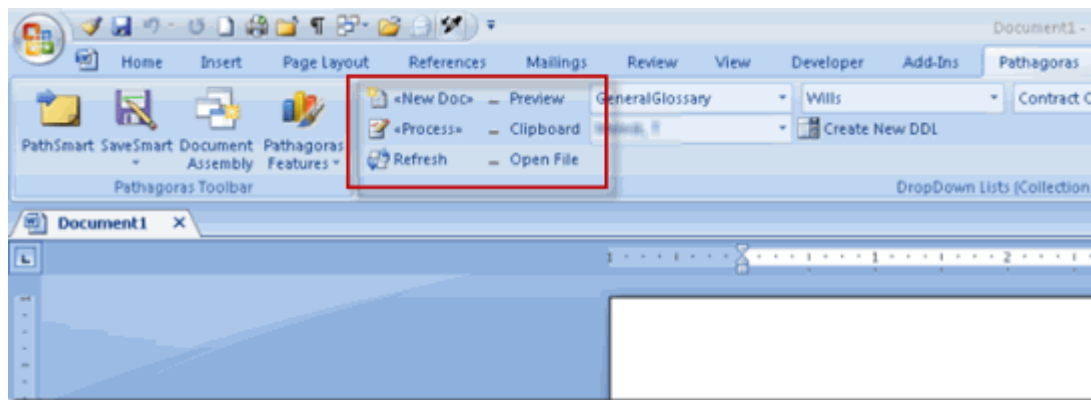
- Pathagoras 2016 added an additional command to the Insert/NewDoc toggle rotation. The new command is "Insert Name". When selected, the name of the document enclosed within << and >> markers will be inserted. In addition to making it easy to copy the name of a document into your current document (instead of having to copy it from other source) it allows you to create Clause Sets of multiple documents. [Clause Sets](#)⁴⁰⁶ are discussed at this link.

17.7 The DropDown List Panel

The DropDown List Panel houses the various DropDown Lists along with the control buttons that allow you to personalize the various Lists to your specific needs.

Note: Before the first list is created, there is only a single button labeled "DropDown Lists". Only after you have created your first list does the panel expand to display the remaining elements.

At the left side of the DropDown List panel are 6 toggle buttons that control how the DropDown Lists (or items you call from a DropDown List) will behave.



«NewDoc/Insert/Insert Name» Toggle

At the upper left of the DropDown List panel, you should see a button showing either «NewDoc» or «Insert» or «Insert Name». The button's title indicates the expected action.

- «New Doc»: a new document will be created, and the selected clause inserted as the first element in the new document; or
- «Insert»: the next selection you make from any list will be inserted into the current document at the current cursor location or
- «Insert Name»: the next selection you make from any list will cause the name of the selection to be inserted into the current document at the current cursor location. You will be given the option to insert the document's short name or its full name (with the full address to its folder location).

The button is a toggle. Click it once to select the 'other' choice.

«Process/Suppress» Toggle:

As you point-and-click in text, Pathagoras default action will be to 'process' any <<*Optional*...>>, <<*Options*...>> and <<*Repeat*>> blocks and to call in any Clause Sets that reside in the inserted text. Automatic processing, however, may on

occasion prove inconvenient, especially if you are testing certain actions and do not want the options text 'touched,' or if you want to call in several terms and you want to delay processing until all are present.

Click the Process/Suppress toggle as needed to control whether processing occurs automatically.


The button's title indicates the expected action.

- **«Process»:** any optional text, and any other text brought in from the source document within <<double angle brackets>> will be processed.
- **«Suppress»:** any optional text, and any other text brought in from the source document within <<double angle brackets>> will be processed, but rather will remain in the assembled document intact and untouched.

The button is a toggle. Click it once to select the 'other' choice.

See Also: [Suppress Processing](#)  158

Refresh: This button should be needed only in the rare situation where you have created a new DropDown List, but Pathagoras has failed to display it. (If you know that a particular List is not up to date, click the «Refresh» button near the bottom of the particular list.)

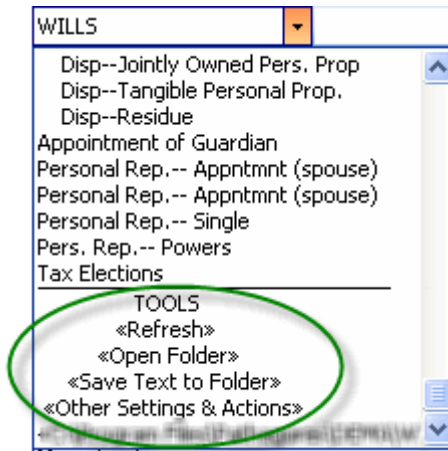
Preview: The classic action when you click on an element in a DropDown List is that it will insert a copy of the selected text at the desired location (into a new document or into an existing document, depending upon the status of the NewDoc/Insert toggle discussed above). But sometimes you want to be able to *preview* the document before committing to it. When 'On' (indicated by a magnifying glass), Pathagoras will display the content of the document you select next in a preview window, along with any comments and usage tips associated with the document. Additional action buttons allow you to insert the text (if it is what you wanted) or to move on to preview another clause. (As noted, the Preview action will show any text that has been placed in the Comments section of the document's Properties section. You should consider adding Comments to your documents to help the end user understand the usage rules for a particular clause. The raw text is nice to see before committing to a selection, but well worded Comments could be more valuable to the end-user. Click here to read more on [Document Comments](#)  670.)

Clipboard: When 'On' (indicated by a clipboard), Pathagoras will send the text of the next item you select from a DropDown List into your system's clipboard. (It's the same action as highlighting text and pressing 'Copy'.) From there you can paste it into the current document, any other document, or any other program (just like any text in your clipboard can be used).

Open File: When 'On' (indicated by the open file icon), Pathagoras will open the original of the next item you select from a DropDown List. (The 'preferred' document assembly action is to display a copy of the text, never the original. But when only the original will do, such as when you want to edit the original text, the Open File toggle can be a real time saver. Just remember that you have the original document open.) (System administrators can 'hide' this Open File toggle. If you do not see it, but want it, ask your system administrator for access.)

Other Controls:

At the foot of each DropDown List is a series of options that make each list even more useful and flexible. Here is what you will see when you scroll to the bottom of a DropDown List:

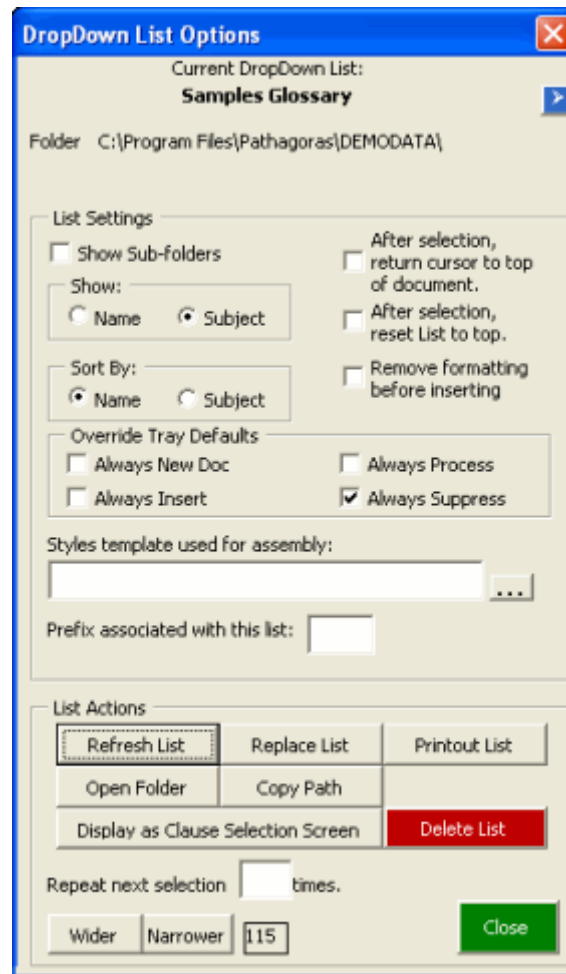


The 'Below the Line' features, Initial Display
 (The very last line, blurred in the sample above, is the full path name
 of the folder or glossary to which the List points.
 Use it as a quick reference to locate the source of the List items.)

- **«Refresh»:** If you have added or deleted items to the folder to which the DropDown List points, you should 'refresh' the List to bring it up to date. It takes just a second or two.
- **«Open Folder»:** Pathagoras knows where everything is. So, if you want to view the contents of the folder to which the DropDown List points, don't navigate to it. Just click this item and you will be taken directly to it. Open documents, edit them, re-save them. All from this one line.
- **«Save Doc To Folder»:** Let's say that you have created (or copied from another source) a section of text (or a complete document) and that you want to save into the folder to which the DropDown List points. Simply click this entry and in no time at all, you can save that text. No navigation. (If the DropDown List points to a glossary, this item will read «Save Text to Glossary». Same idea. Same ease of use.)

See [Adding Content to DropDown List](#) ³⁵⁹.

- **«Other Settings & Actions».** This screen first appears in Pathagoras v. 2013. When clicked it displays an extensive list of options that control the look and feel of the particular DropDown List.



The 'Other Settings & Actions' Display

List Settings.


- Show Sub-folders: Turn the Tree Service 'on' and the DropDown List display links to the sub-folders beneath the parent. (Sub-folders are displayed at the top of the list between curly-braces.) Click on a sub-folder and the list will be re-drawn, containing the files of the selected sub-folder. Any sub-sub-folders will be displayed at this level as well, along with an "{..{ Up} }" entry so that you can return to top of the tree.
 ➔ The folder display possibilities here are quite literally endless. By strategically assigning parent folders, you could conceivably access every folder and sub-folder without ever leaving your editing screen. All navigation as envisioned by Windows could be eliminated.
- Display Names/ Display Subjects: Click as appropriate.
- Sort by Name/Sort by Subject: Click as appropriate.
- List Reset after Insert / No Reset after Insert: This simply indicates what shows after you click on a clause in the list and the list is 'retracted'. If 'Reset' is chosen, the title of the list displays. If 'No Reset' is chosen, the selected item displays, making it a bit easier to (perhaps) choose an item further down in the list.

- Return Cursor to Top of Document / Leave Cursor at End:
 - ✓ Choose the former if you want the program to reset itself to the top of the document (so that you can easily review the document from top to bottom)
 - ✓ Choose the latter if you want the insertion point (and the display) to be at the document bottom so you can easily add additional text.
- Remove formatting: Sometimes you don't want the clause you are about to insert to contain the formatting with which it otherwise has been saved. Here is the way to insert truly unformatted text.

Toggle Overrides.

In the DropDown List area of the editing screen, there are two toggle buttons that control whether the selection will cause a new document to be created (vs. inserting the selection into the current document and whether the <<Options/Optional/Repeat>> blocks (if any) within the inserted text will be processed or (temporarily) ignored. If you know that the clauses in a certain DropDown List should always be handled in a particular fashion, you can set that in this section. Regardless of the toggle button setting, the setting in the individual DropDown List will be honored.

Other Settings:

- Assign a Template: Assign a template to any documents created from a term in the list. A template is typically a blank document that contains headers, footers, margins and styles unique to the type of document contained in the list). Once a template is assigned, if to be laid down before any text is inserted in the following instances: (1) the New Doc toggle has been selected and (2) when the item called from the list is the very first item on an otherwise blank page. (Irrespective of the assigned template, if you call in a term into an otherwise blank document that itself has headers and footers, the headers and footers of the recalled document will display.)
 - Refresh List: If you have added more items to the folder or glossary, refresh the List to include the new additions. It takes just a second or two.
 - Replace List: Place another, completely different, folder or book in place of this DropDown List.
 - Open Folder: Don't navigate to the folder to see its full contents. Just click this item and you will be taken directly to it. Open documents, edit them, re-save them. All from this one line.
 - Copy Path: Simply places the name of the folder to which the DropDown List points into your clipboard memory. Handy when you want to navigate to the folder. (But don't forget about the 'Open Folder' button.
 - Display as Clause Selection Screen: Create a Clause Selection Screen from the entries in the DropDown List. Very helpful when you want to quickly select and assemble multiple items in the list. Imagine—document assembly of image files. Or Word Perfect® files!
-  If you want to create a sub-set of this list, display the list as a Clause Selection Screen. Then select just the clauses you want to appear in your DropDown List. Choose the Create DropDown List option and press Next. Instantly, you have a shorter DropDown list containing a hand picked selection of terms.

Miscellaneous

- Create Printout: Need a hard copy of the contents of the DropDown list? Click this entry to transfer the information to a Word document for editing/printing.
- Delete: Delete the DropDown List from the display.
- Wider & Narrower: (Word 2003 only) Adjust the width of the display with these Lists.
- List information: If you click either the very *first* or the very *last* entry in the list, you will see a page of information regarding the list. It may be helpful in determining any unusual behavior of the list.

17.8 Collections

Pathagoras lets you create and maintain up to 10 DropDown Lists simultaneously. But, if yours is a multi-discipline practice, or your clause lists are expansive, even ten may not be enough. Examples:

- Let's say you have an active Estate Planning practice. In order to speed up your access to clause and documents on this subject, you have created DropDown Lists for (1) Wills (2) Trusts (3) POAs (4) Living Wills (5) Probate and (6) Letters. That leaves you with only 4 more Lists for other subjects.
- Let's further say that you also have an active Domestic Relations Practice. You would like DropDown Lists for (1) Pleadings, (2) Discovery, (3) Separation Agreements, (4) Orders and (5) Letters. But now there is not enough room!
- And what about your Personal Injury or Real estate practice? All your DropDown Lists are consumed by other topics.

Pathagoras 2014 new feature:

You can now create easily switchable 'Collections' of DropDown Lists. Pathagoras allows you to create and save up to 4 such collections, each on any topic of your choice. (A 5th collection exists, but we have reserved it for use by the administrator. It is hidden in the background from 'non-authors.')

Here is a sample of DropDown List collections a law firm might have.

Estate Planning		Domestic Relations	
(1) Wills	(6) Letters to Clients	(1) Separation Agreements	(6) Letters to Clients
(2) Trusts	(7) Letters to Court	(2) Pleadings (documents)	(7) Letters to Court.
(3) POAs	(8) Variables	(3) Pleadings (clauses)	(8) Variables
(4) Living Wills		(4) Discovery	
(5) Probate		(5) Orders and Decrees	
Real Estate		Office Admin	
(1) Deeds	(5) Letters to Clients	(1) Retainer Agreements	(6) Seasonal Correspondence
(2) Deeds of Trust	(6) Letters to Court	(2) Letterhead options	(7) Variables
(3) Sales Contracts	(7) Letters to Financial	(3) Letters to Clients (body text)	
		(4) Signature blocks	

(4) Leases	Institutions	(5) Fax Cover sheets
(5) Closing Forms	(8) Variables	

So when you want to work on a divorce action, call up the Domestic Relations collection and all the forms on that topic are now 2 clicks away.

Need to quickly prepare a Will. Not a problem. Call up the Estate Planning collection.

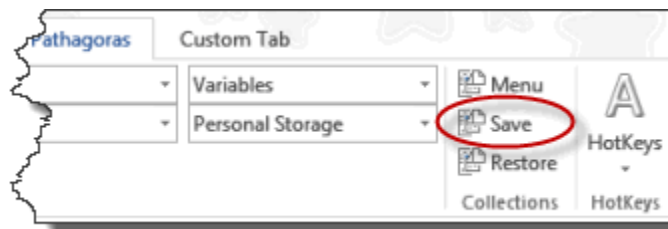
The collections switch in and out in a matter of 2 or 3 seconds!

The above chart reflects possibilities only. Every name, every topic, every decision is yours. The possibilities are endless. The flexibility is unlimited. (Note that the list "Variables" appears in every list. There are a few 'community' lists that probably should be included in every Collection. See [this section](#)³⁷⁵ of the Manual for more information.

To Save a Collection:

Of course, the first objective is to create a set of DropDown Lists that reflect the documents and clauses to which you wish two-click access. Creating your DropDown Lists is the subject of the immediately preceding sections of this Manual.

Once you have set up an initial group of DropDowns that you like, click the Save button in the small cluster of buttons in the menu area of your screen labeled 'Collections'.



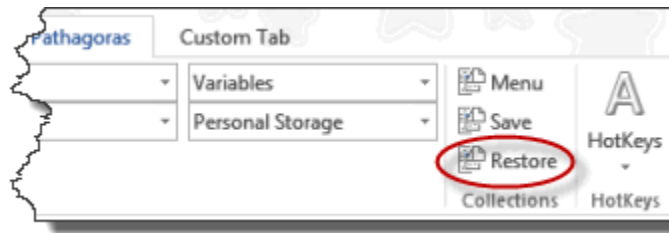
Click 'Save' to display the Save Collections Menu

With the 'Manage Collections' screen visible, select the number (1 thru 4) of the slot into which you wish to save the collection. If you are saving a new Collection on a 'blank slate', it doesn't matter which slot you pick. If you are updating or overwriting an existing Collection, select the slot you wish to overwrite. (Pathagoras will display in red outline the last Collection you saved or recalled.)

You can add a title for the Collection by clicking the Rename button at the foot of each collection. (If you wish to establish a Document Assembly 'Environment' (discussed below), make sure that the name of your Collection and the name of the document assembly Library are the same.)

To Restore a Collection:

Manually: Same steps as above, but instead of clicking 'Save', click the 'Restore' button in the Pathagoras menu area.



Click 'Restore' to display the Restore Collections Menu

When the Restore Collections screen is displayed, click the Restore button beneath the desired Collection.

Automatically: If the name of the Collection matches the name of a Library, and you select a Library from the Document Assembly (Libraries & Books) screen, the 'Restore Collection' routine will happen automatically. (You will, however, be asked if you want to restore the Collection. Just say 'Yes'.)

Document Assembly "Environments":

"It would be 'really cool' if, when I switch Libraries (e.g., from Domestic Relations to Estate Planning) that not only do I get a new set of books to build documents from, but a new set of DropDown Lists."

You now can! See the next section discussing [Document Assembly Environments](#). 371

17.9 Environments

"It would be 'really cool' if, when I switch Libraries (e.g., from Domestic Relations to Estate Planning), that not only do I get a new set of books from which I can build documents from, but a new set of DropDown Lists to use to augment those documents, and to take care of ancillary business." -- a satisfied customer with a great new idea

Introducing Document Assembly Environments:


DropDown Lists and Libraries normally work independently of each other. However, if you give a Collection a name that is identical to that of an existing Library, the two can change in tandem with one another.

The result is a true Document Assembly 'environment,' allowing you instant access to all books of the Library and to all documents reflected in all DropDown Lists in the Collection of the same name. With Environments, you can build complex documents via the Clause Selection screens, and then instantly insert additional text from DropDown Lists.

These new Collections allow you to create operational 'environments'.

To create an environment, you would follow the same steps outlined in the [Collections](#) 369 pages of this Manual.

The 'environment' is automatically enabled when the name you assign to the Collection matches the name you have assigned to a Library.

 An 'Environment' call is triggered from the Document Assembly Libraries. It is only one way -- a selection of a new Library can trigger the restoration of a Collection of the same name, but the reverse is not possible.

17.10 DDL Variables List

DropDown Lists provide two-click access to a variety of items: documents, clauses, text snippets (a/k/a building blocks), Excel data, PowerPoint presentations, image files, etc. The target end-product is a signable letter, contract, pleading, Will, etc. The target audience is typically the client or customer.

But 'source-document authors' (here we refer to the author of the original or 'source text' used to create the client documents) can also benefit from DropDowns. Several authoring tools can be presented via DropDown Lists. We discuss in this section the DropDown Variables List. This List gives you two-click access to the variables you have adopted as office standards for your forms. It can dramatically speed up the process of 'Pathagorizing' your documents.

It is easy to create, maintain and use the DropDown Variables List. This List is 'permanent' (at least until you manually delete it) sits at the top of your editing screen along with other DropDown Lists. It is 'always on' and 'always active.' (You can create more temporary lists of variables using your collection of masks. These temporary lists are discussed [at this page of the Manual](#) ¹²⁹.)

Creating the DropDown Variables List

Two methods are available:

1. Via the Instant Database Screen"

- Display the Instant Database screen and select a mask or a client/customer's record that contains the variables you want to add to the Variables List.
- Click 'Power Tools' (the red button) and then click the "Insert into DropDown List" button. If a List does not already exist, Pathagoras will ask for the location (row and slot) where the List should be placed.
- If a list already exists, you will also have the option of replacing the existing List with just the variables in the mask or record you have called, or to add any new variables to the existing List.

2. Via the 'Collections' Menu (ribbon)

- Click the Pathagoras tab and locate the 'Collections' section toward the right of the ribbon.
- Click the 'Menu' button.
- Select 'Variables' toward the bottom of the new screen. Pathagoras will add a new DropDown List called (appropriately enough) 'Variables.' (If a list of variables already exists from your having followed these steps in an earlier setting, those variables will populate the new list. Otherwise, Pathagoras will create a new list of variables. It will insert [Client Name], [Client Address], and [Client City, ST ZIP] as a starter list. These variables can be kept or deleted following the Editing procedures below.

Insert Variables in 2 clicks!

- **Single insertion:** Just like any other DropDown List, it's point and click. Just make sure that the cursor is at the location where you wish the variable to be inserted.
- **Global replacements:** Let say you have on your editing screen a document that you want to 'Pathagorize.' Currently it has the name 'John Doe' in several locations throughout the document. You want to 'neuter' the document to create an office form.
 1. Highlight one instance of a word or phrase (i.e., John Doe) that you want to convert to a variable.
 2. Click on an entry in the Variables DropDown List that you want John Doe to become.
 3. Pathagoras will ask if you want to replace all other instances of the same text in the document with the selected variable. If you do, click 'Yes.' The replacement of all 'John Does' with the selected variable is done in a split second.
 4. As with all replacements performed by Pathagoras, the program will preserve the style of the text --case, bold, italics, color, etc. -- when replacements are made in the other locations.

This global replacement of personal values with variables is almost identical to the action that can be performed using the ['Create Variables' Wizard](#)¹²² (Alt-V). The main difference is that, with the DropDown Variables List, the names of your variables are always at your fingertips.

Adding Variables to the Variables List

- **Highlight and Add:** A single variable, or a group of variables, that currently reside on you editing screen can be quickly added to your List.
 1. Highlight the variable(s). Click on the Variables list and select the 'Add Variables to List' element at the bottom.
 2. To add more than one variable at a time, type a list of the variables in a stack down the left side of the page. Highlight the stack and the click the 'Add Variables to List' element at the bottom.
 3. If you are editing a document that contains variables you wish to add, but they are not stacked, use the 'Via the IDB Screen' method described in the section that immediately follows. See #3 in that section.)
 4. That's it. Your new variables have been added to the bottom.
- **Via the Instant Database (IDB) screen:**
 1. Display the Instant Database screen. (You can press Alt-D to call up the IDB screen).
 2. Select an existing record or any mask from the lists in the upper right or upper left drop downs.
 3. As an alternative to using an existing record, you can click the <Scan> button to place the variables in the currently open document into the IDB screen.
 4. Regardless of how you populated the IDB screen, click the red 'Power Tools' button.
 5. Click the 'Insert Into DropDown List' button. Choose whether you want to

replace the existing List, or add any new variables to the List.

6. All done.

- **Manual Add:**

Pathagoras allows you easy access to the file that contains the Variables. Using the steps outlined below under "Editing the Variables List" you can manually insert (remove, rename, etc.) variables at your pleasure.

Editing the Variables List.

- The variables in the Variables List are stored in a file called Variables.txt. This file is located in the same folder that your other Instant Database records are stored. By default, the location of the Variables.txt file is:

32 bit computers: c:\program files (x86)\Pathagoras\IDBS

64 bit computers: c:\program files\Pathagoras\IDBs.

- You can easily access the file for editing in these two ways:

1. **Via the Variables List:** Click the 'Open List for Editing' element near the bottom of the List itself.
2. **Via the Collections menu:** Shift click on the 'Variables' button.

Pathagoras will open the document called Variables.txt as a Notepad (not Word) file. (It is just easier to edit a simple file such a Variables.txt with this editor.) Edit as appropriate and then save and close the file.

If you need a more powerful editor (i.e., Word) to edit the file (because let's say you want to search for and replace text in a large file, or you want to alphabetize your entries), you can easily copy all or part of the content from the Notepad screen and paste it into Word . Edit as desired. Copy and paste the results back into the Notepad screen, save and close.

Don't forget to click «Refresh» in the Variables List.

(You can also navigate to and edit the Variables.txt file manually. As suggested above, we recommend that you edit the list with Notepad, Microsoft's simple text editor.)

Currently there is only one Variables List available. If you wish to categorize the List (i.e., separate your Estate Planning variables from your Litigation Variables, etc.), you can do so by editing and rearranging the contents of Variables.txt using any of the editing techniques discussed above. You can insert Category captions (perhaps in ALL CAPS). You can even indent the variables within each category for a more visually pleasing display.

E.g.,
GENERAL
 [Client Name]
 [Client Address]
ESTATE PLANNING

[Child@1 Name]
[Child@2 Name]
REAL ESTATE
[Grantor]

Refreshing the List

If you know you have added a variable (or several) to the List, but it doesn't appear when you drop it down, click the «Refresh» entry near the bottom of the List. This tells Pathagoras to reread the Variables.txt file.

Variables List and the Quick Access Toolbar (QAT)

If you frequently need to add variables to your documents, you should consider elevating your Variables DropDown List to your Quick Access Toolbar (QAT). That way you will have immediate access to your variables without having to first click on the Pathagoras tab.

To add the List to the QAT, right click on the List and select 'Add to Quick Access Toolbar.'

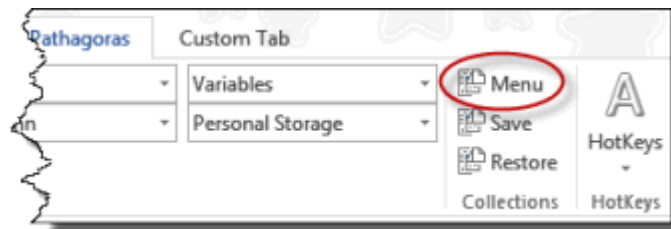
If you have placed your Variables DropDown in more than one Collection, make sure that the List is in the same relative location in each. If you change Collections, the QAT will display the DropDown that occupies the same position as the List from which you created the QAT entry.

17.11 Community DropDown Lists

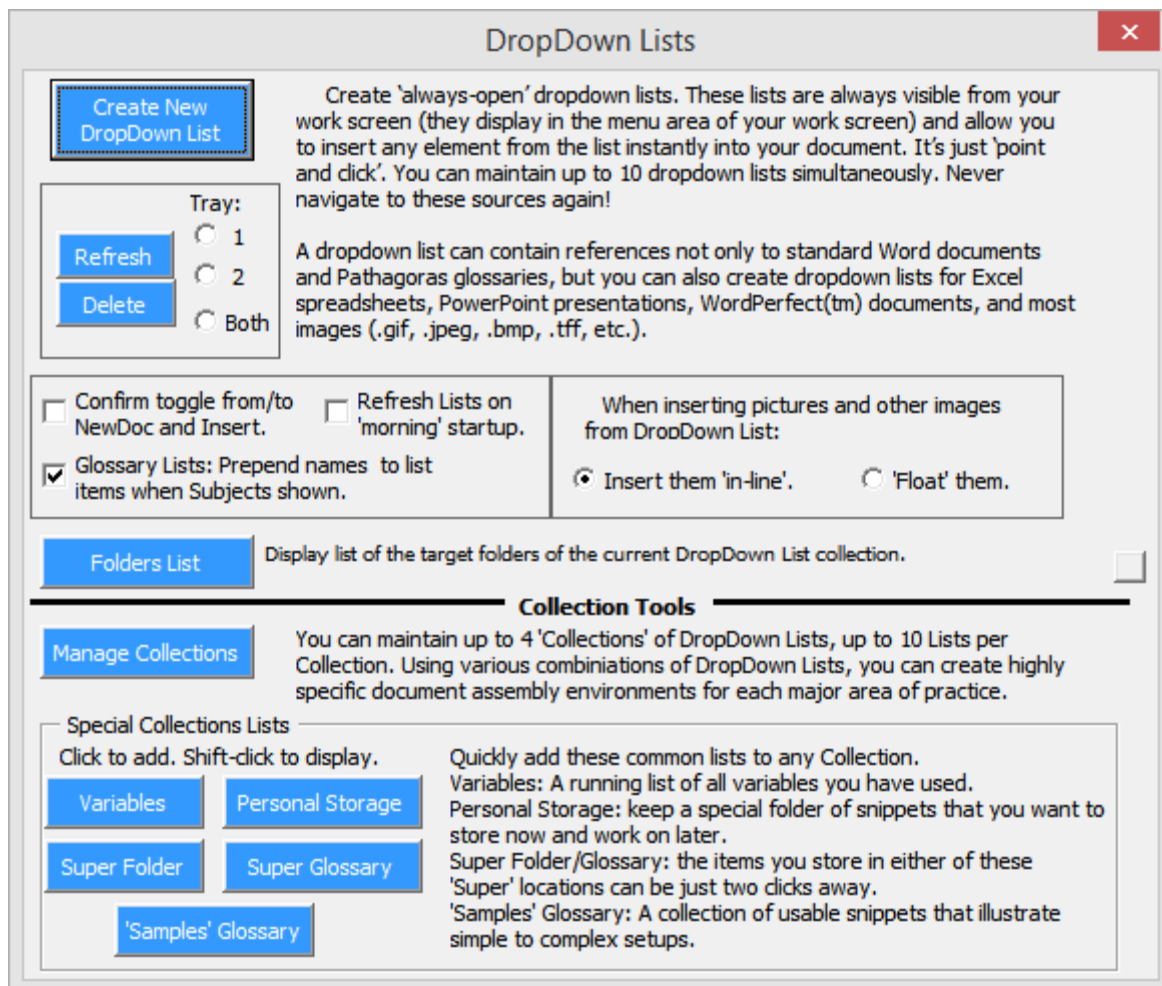
Pathagoras offer several other 'special' lists that you may want to add to one or more (or all) of your collections. We call these 'Community DropDown Lists.' Their content is more general in nature (cover sheets, signature blocks, notary jurats, pleading styles, and the like.) Because of their general content, it is appropriate that you consider including them in multiple Collections. Pathagoras makes it easy to do so. These community lists include:

- You **Super Folder** and **Super Glossary**
- **Samples Glossary**
- **Your Variables.** (if displayed you can insert any variable that you have save in the List into your document in just two clicks. No more guessing how you spelled the variable. The previous page is dedicated to the DropDown Variables list.)
- **Personal Storage**

You can manually navigate to any of these lists, but Pathagoras has made access to them much easier via the Collections Menu (visible in the Pathagoras toolbar area). Click the Menu element.



Click Menu to display the DropDown List Menu, shown below



The Special Collections Lists are shown at the bottom.

Super Folder⁹⁰: The SuperFolder is a folder typically containing your 'general' documents. It and its sibling SuperGlossary, below, allow you quick, no navigation access to the content you store there. If you have assigned a SuperFolder, it's a good idea (but not mandatory) to reference it with a DropDown List. (If you have not previously assigned your Super Books, Pathagoras will give you that opportunity. To learn more about the benefits of **SuperBooks**⁹⁰, click the link.)

[Super Glossary](#)^[91]: The SuperGlossary is a glossary typically containing 'general' clauses. It and its sibling SuperFolder, above, allow you quick, no navigation access to the content you store there. If you have assigned a SuperGlossary, it's a good idea (but not mandatory) to reference it with a DropDown List. (If you have not previously assigned your Super Books, Pathagoras will give you that opportunity. To learn more about the benefits of [SuperBooks](#)^[90], click the link.)

[DropDown Variables](#)^[372]: This is discussed fully in another section of this Manual.

Personal Storage: This list points to a Windows folder into which you can temporarily place documents, clauses, building blocks and text snippets that you wish to preserve, but don't quite know where the ultimately belong. So save them now and process them later. Consider this just a temporary storage location.

17.12 Debugging the List

There are two main reasons why the entries in the DropDown List may not appear as you expect them to:

- **Issue:** "I know that a file is in the folder (or clause is in the glossary) to which the List points. But that file or clause does not appear in the DropDown List. Why not?"
 - **Solution:** The List just needs to be refreshed. Click the «Refresh» entry near the bottom of the DropDown List
- **Issue:** "My DropDown List is set to display the subjects of the terms in my glossary, and I have assigned subjects to all of my terms. However, for several entries in the List, the term '(no subject)' appears. Why?"
 - **Solution:** This issue has been reported only when a 'glossary' is the object of the DropDown List. The cause is that there is an extra line between the Subject (the blue text in the glossary just above the actual term text) and the beginning of the actual term text. (For the glossary to be properly read, there can be no lines between the red name, the blue subject and the beginning of the actual text of the term.)

You can fix this by displaying the glossary and *manually* removing the extra line.

--Or--

You can run Pathagoras' '*Structure Checker*' against the glossary. This is the better alternative since it will not only fix the problem at hand, but it will check the entire glossary structure for you.

Here are the steps to run the Structure Checker:

1. Display the glossary. (The quickest way is to click the «Open Glossary» item from the DropDown List or Open Glossary from the Document Assembly 'Libraries & Books' screen.)
2. With the glossary 'on display,' click the Pathagoras dropdown features list and select "Authoring/Editing Creation Tools".

3. Click "Glossary Tools".
4. Click the <Structure & Integrity> button.

17.13 WordPerfect, PDF and Image Assembly

Word Perfect Assembly

Pathagoras is capable of 'document assembly' of WordPerfect® files assigned to a DropDown List. We do want you to understand however, that the resulting document is a Word, not a WordPerfect document.

Individual files can be 'clicked in' one at a time using just the List.

However, true assembly using these WordPerfect documents can be accomplished by clicking the <<Clause Sel. Screen>> entry toward the bottom of the DropDown List. Pathagoras will transfer all elements of the List into a standard Clause Selection Screen, with the selectable items will be listed in the left panel.

Just like in assembly of standard Word documents, you would select some or all of the WordPerfect documents and move them to the right panel. When done, press **Next>>**.

See Also:

[WordPerfect®](#)  605

[Folder to Glossary](#)  318

WordPerfect® is a registered trademark of
the Corel Software Corporation

PDF Assembly

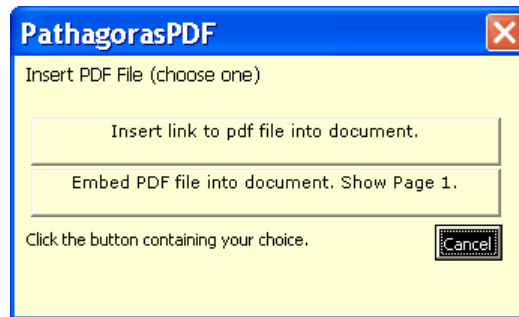
As mentioned in other pages, Pathagoras allows a DropDown List to be populated with PDF files as well as document files. Therefore, Pathagoras is capable of 'PDF assembly' as well as document assembly.

There are a few limitations of which you need to be aware when it comes to working with PDFs.

When you click on a PDF from a DropDown List, you will be presented the option of inserting a link to the file or the actual file. See below figure.

Link to PDF: A link is no more than that. Make sure that the recipient of the link has access to the folder to which the link points.

Embed PDF: The other choice is to actually embed the PDF in the Word document. This makes for true document portability. It can also make for a very large file, as the Word document now contains not only the Word text but the entirety of the PDF record. You can embed ('assemble') an unlimited number of PDF files into your Word document.



Viewing the embedded PDF: When you click the embed option, an 'image' of the first page of the PDF file will be placed into your Word document. To view the actual file, simply double click on the image. Your PDF viewer will activate and show all of the PDF.

Printing the PDF: Word cannot directly print an embedded PDF. If you printed a document into which you embedded a PDF file, you will get all of the Word text and the first page (the one that you see when you embed the file). To print the entire contents of the PDF, you must view it (see above paragraph) and print using your PDF viewer.

Converting Word files to PDF: You cannot convert your Word document into a PDF and keep the embedded PDFs intact. If you convert your Word document containing embedded PDF files into a new PDF file, only the *image* of the first page of each embedded file will convert. If each PDF file is itself one page, this is not bad news, but it is little different from image assembly. If you need to transmit the Word document with the embedded PDFs as a single entity, don't convert it to PDF.

See also:

[Creating a DropDown List \(Free Hand\)](#)  358

IMAGE ASSEMBLY:

As mentioned in other pages, Pathagoras allows a DropDown List to be populated with image files as well as document files. Therefore, Pathagoras is capable of 'image assembly' as well as document assembly.

Images can be 'clicked in' one at a time, perfect for situations where an image is inserted and a description is typed after each entry.

True image assembly can be accomplished by clicking the <<Clause Sel. Screen>> entry toward the bottom of the DropDown List. Pathagoras will transfer all elements of the List into a standard Clause Selection Screen, with the selectable items listed in the left panel. Select some or all of the images and press **Next>>**.

The Pathagoras System

'Repeat' Blocks

Part

XVIII

18 'Repeat' Blocks

The 'Repeat' function allows you to duplicate a particular block of text a designated number of times.

When a <<*Repeat*(text)>> prompt is encountered, Pathagoras will bring in a second (third, fourth, etc.) copy of the text within the scope of the prompt 'X' number of times.


Pathagoras can provide this 'repeat' feature in a wide variety of ways. They are summarized here, and elaborated upon, with examples, in the subsequent sections of this Manual:

Simple: In its simplest iteration, the structure of a 'repeat' block is simply "<<*Repeat *(text)>>" (with (text)being anything you wish: text, a signature line, a variable, anything.

Simple with incrementing variables. If you include bracketed variables within the scope of the <<*Repeat*>> prompt, Pathagoras will add a suffix to each repeated of the variable and increment its 'name' by one. "[Name@1] [Name@2] [Name@3], etc.

!Groups! You can add an optional !groupname! to the key text to link the 'repeats' value assigned when the first member of the repeat !group! is encounters to the repeats blocks in other locations in the document

Series Connectors: Pathagoras can add commas and an appropriate conjunction so that the repeated text can be made into a grammatically correct list. ("A1, A2 and A3" instead of simply "A1 A2 A3")


 You can reference an external clause (document or glossary term) instead of 'hard coding' in the text. The clause must either be one named with the [prefix/suffix](#)⁴²⁰ naming style (and with the prefix properly registered), be 'fully qualified (full path name) or located in one of the search locations listed in the [Order of Search](#)⁹⁵ page.

=====

But wait! There is still more!



After you have run any of the above routines and brought in, let's say, 5 'repeats' of a particular clause, and you find that you need a sixth, type the word "repeat" at the location where you need the next clause, followed by the key press <Alt-G>.

 Proper construction of the <<*Repeat*>> block is a must. Use the 'Structure Checker' found under [Editing Tools](#)⁴⁴⁶ to make sure that the '<<' and '>>' counts balance.

See Also:

['Repeat' Settings](#)³⁹¹

['Repeat' in Clause Selection Screen and DropDown Lists](#)³⁹⁵

['Repeat' Alternatives](#)³⁹⁵

18.1 Simple Repeats

The structure of a simple <<*Repeat*>> block is simply the introductory word "<<*Repeat *" followed by whatever text you want to repeat, followed by a closing ">>"

Example (The colors are added for illustration purposes. They are *not* required):

```
<<*Repeat *Row, row, row your boat, gently down the stream.>>
```

When this above is encountered during a document assembly session, Pathagoras will pause and ask "How many times do you want to repeat this text?". If, let's say, your answer to the question is 3, the 'return' of the function will be

```
Row, row, row your boat, gently down the stream. Row, row, row your
boat, gently down the stream. Row, row, row your boat, gently down the
stream.
```

If you add an 'Enter' just before the closing '>>', i.e.,

```
<<*Repeat *Row, row, row your boat, gently down the stream.
>>
```

the processed text will appear like this:

```
Row, row, row your boat, gently down the stream.
Row, row, row your boat, gently down the stream.
Row, row, row your boat, gently down the stream.
```

18.2 Simple, with incrementing variable

The above causes a repeat of the identical text X number of times. You will more likely will want to use the <<*Repeat*>> function to increment a variable that you place within the repeating text. The structure is identical to the 'Simple' example above, with the introductory word "<<*Repeat *" followed by whatever text you want to repeat, followed by a closing ">>"

Examples (The colors are added for illustration purposes. They are *not* required):

```
<<*Repeat *[ChildName]>>
```

When the repeat block is encountered during a document assembly session, Pathagoras will ask "How many times do you want to repeat this text?" The 'return' of the function would be, in this case, the named variable, incremented by the number of repeats called for. E.g, [ChildName@1] [ChildName@2] etc

You can include as much or as little additional text within the repeat block as you wish. All plain text will be repeated 'intact'; all bracketed text will be incremented as per the examples.

You can increment a number outside of a variable using the character set '[#]' (no quotes).

For example, if you wanted to make a numbered list using the above, you might type:

```
<<*Repeat *[#]. [ChildName].
>>
```

Suppress incrementing:

By default, all bracketed variables are incremented by each repeat called for. However, if you place a variable that you do not wish to increment (perhaps a date variable that will be consistent for all blocks, simply append an underline to the variable name (inside the bracket) and Pathagoras will repeat the variable but not increment it.

Example:

```
Signatures:
<<*Repeat *_____ [Date_]
[Buyer]
```

>>

You can modify the settings of the 'non-increment' function, including the non-increment ending character and whether it is removed (let's say so that it will match other variable) or left in. "_" and 'remove' are the default settings.

The settings are found under the "Repeats" tab of the Utilities/Settings | All Settings screen.

18.3 Repeat with !Groups!

Frequently there is a connection between a <<*Repeat*>> prompt at the 'top' of a document with elements further down. For example if there are 'three' shareholder accounted for at the top of the document, three additional text blocks may be needed further down in the document to account for the number of shares each holds. The various <<*Repeat*>> blocks can be tied together with a common !groupname!

Structure: <<*Repeat*!groupname!*(text1)>> . . . <<*Repeat*!groupname!*(text2)>>.

When encountered during a document assembly session, Pathagoras will ask "How many times do you want to repeat (groupname)?"

Example (A 'repeat ask' is placed at the top of the document. It's sole purpose is to ask 'How many shareholders?' If the repeat ask is not included, the 'How many shareholders?' question will be asked when Pathagoras encounters the first substantive 'repeat block'):

```
<<*Repeat*!shareholders!*>>
```

The following individuals own stock in the corporation:

<u>Name</u>	<u>Shares Owned</u>
<<*Repeat*!shareholders!* [ShareholderName], [SharesOwned]	
>>	
(. . .)	
The shareholders waive notification of the upcoming annual meeting as evidenced by their signatures below:	
<<*Repeat*!shareholders!* _____	
[ShareholderName]	
>>	

When encountered during a document assembly session, the very first line (the repeat gatherer) causes Pathagoras to ask for the number of shareholders. It will use the answer further down the document to duplicate the text between the remaining << and >> markers the designated number of times. Of course, all [variables] inside any repeat block will be properly incremented.

Here the result if the answer to "How many shareholders? is 3.:

The following individuals own stock in the corporation:


<u>Name</u>	<u>Shares Owned</u>
[ShareholderName@1],	[SharesOwned@1]
[ShareholderName@2],	[SharesOwned@2]
[ShareholderName@3],	[SharesOwned@3]
(. . .)	

The shareholders waive notification of the upcoming annual meeting as evidenced by their signatures below:

[ShareholderName@1]

[ShareholderName@2]

[ShareholderName@3]

 The !GroupName! that you assign to a Repeat block can be shared with subsequent <<*Options*>> blocks carrying the same name. Some restrictions apply. If you want the Repeat value to controls a subsequent Option,

1. You must set the Repeat Value first. It must be set via an <<*AskRepeat*> question at the top of the document
2. The number of choices you can set for the <<*Options*>> block can be 3 and only 3. While the actual text can be anything, the nature of the text must be as follows:
 - The first choice must reflect the text you want in the document if the Repeat value is 0.
 - The second choice must reflect the text you want in the document if the Repeat value is 1.
 - The third choice must reflect the text you want in the document if the Repeat value is 2 or more.
 - The 'reason' for this rule is not necessarily obvious, so here is our stab at an explanation:

The primary interplay of a Repeat value to a subsequent Options block is to set noun to verb consistency. In English, 'None *are*', but 'One *is*' and 'Two (and three, four, etc.) *are*.' Plus, words like "in equal shares" would apply to an estate distribution to two or more children, but not to 0 children or to one child. Other examples abound that suggest that 3 choices can address the full range of possibilities.

Further, it simply would not be reasonable (or practical) for a document author, faced with a possible repeat of 100 shareholders, would have to compose 100 separate choices in an Options block.

- Our surveys suggested that 3 is the right number.


Example:


```
<<*AskRepeat*!Children!*>>
```

```
<<*Options*!Children!*There were no children born of our marriage./One child,
namely [ChildName], was born of the marriage/There were
```

```
<<*Repeat*!children!Return*one>> children were born of the marriage and their names
are <<*Repeat(and)*!children![Child]>>>>
```

3. Not so much a rule, but strong advice: Don't accidentally reset the Repeat value by creating an *AskOptions* or *AskOptional* prompt with the same groupname. This will cause the Repeat value to be overwritten.
4. You can avoid the above restrictions simply by having an Options !groupname! that is different from the Repeats !groupname!.

 The use of !Groups! is found in other Pathagoras routines. See [Variables\(!Groups!\)](#)¹¹⁹ and [Options\(!Groups!\)](#)¹⁵¹

 If you want to return to the editing screen the spelled out value of a number that has previously been assigned to a !groupname! for a Repeat, you can do so by enclosing the !groupname! within double angle brackets. E.g., from the above examples, <<!**shareholders**!>> will return 'three' to the screen if '3' was the answer to the 'How many shareholders are there' prompt higher up in the document.

18.4 Anatomy of the Repeats Block

```
<<*Repeat*!children!Return*one>> children were born of the marriage and their names are
<<*Repeat(and)*!children!*[Child]>>
```

Anatomy: Two sets of 'facing' angle brackets (the **boundary** characters) surround the core block. A groupname is used to tie two or more Repeat blocks together. The 'Return' call tells Pathagoras simple to 'return' the response requested when the repeat block is processed to the screen 'intact'. The administrative section is closed by a third asterisk. (The 'one' indicates the 'style' of the return. In this example, the number will be typed out instead of it being the actual digit. In the second Repeat block, the "(and)" tells Pathagoras to use commas as separators between a series of two or more repeated variables, the group name !children! ties back to the first Repeat block to obtain the initially set value. The administrative section is closed by a third asterisk. [Child] is the variable that will be repeated 'X' number of times, each time incremented.

18.5 Series Connectors ('and' or 'or')

In the previous examples, the text was repeated 'intact' with no embellishment except the incremented variables.

However, you may wish your 'repeated' text to appear in the final version of your document in a 'series' style display, with commas and a conjunction in the appropriate location, e.g., "[child@1], [child@2], [child@3] and [child@4]."

To accomplish this, you will want to add a 'series connector' to the <<*Repeat*>> prompt. It is a very easy step.

There are only two possible series connectors: "(and)" and "(or)". Just type the desired

connect "(and)" or "(or)" immediately after the '<<*Repeat' keyword.

Examples:

No series connector:

```
<<*Repeat*[ChildName]>>
```

Result, if "3" repeats chosen:

[ChildName@1] [ChildName@2] [ChildName@3]

With a series connector:

```
<<*Repeat(and)*[ChildName]>>
```

Result, if "3" repeats chosen:

[ChildName@1], [ChildName@2] and [ChildName@3]

With a series connector and a group:

```
<<*Repeat(and)*!NumChildren!*[ChildName], [ChildDOB],>>
```

and the same variable(s) elsewhere in document:

```
<<*Repeat(and)*!NumChildren!*[ChildName]>>
```

Result, if "3" repeats chosen:

[ChildName@1], [ChildDOB@1], [ChildName@2], [ChildDOB@2], and
[ChildName@3], [ChildDOB@3],

and elsewhere in document:

[ChildName@1], [ChildName@2] and [ChildName@3]

Note: With series connectors, not only is the conjunction 'and' or 'or' added, but the appropriate commas separating the initial elements in the listing are added.)

Connector punctuation:

If you want to specify the punctuation between elements, you can specify it after the 'and' or 'or' connector. E.g.,

```
<<*Repeat(and;)*!NumChildren!*[ChildName], [ChildDOB]>>
```

Result, if "4" repeats chosen:

[ChildName@1], [ChildDOB@1]; [ChildName@2], [ChildDOB@2];
[ChildName@3], [ChildDOB@3] and [ChildName@4], [ChildDOB@4]

18.6 Repeating Tables & Rows

Tables:

- If you wish to repeat the contents of an entire table, surround the table with a '<<*Repeat* block>>'. E.g.:

<<*Repeat*!Shareholders!*

[Shareholder]	[SharesOwned]
[ShareholderAddress]	

>>

- If the table contains a header row or rows, you can tell Pathagoras to automatically repeat just the non-headers rows. Here's an example (similar to the above):

<<*Repeat(header)*

In the following section, provide the requested information regarding each child:		
Name of child	Address	Date of Birth
[Child]	[Childaddress]	[Date of Birth]

>>

When you “Process Page” on the above example, Pathagoras will ask for the # of repeats. If Answer=3, the table becomes:

In the following section, provide the requested information regarding each child:		
Name of child	Address	Date of Birth
[Child@1]	[Childaddress@1]	[Date of Birth@1]
[Child@2]	[Childaddress@2]	[Date of Birth@2]
[Child@3]	[Childaddress@3]	[Date of Birth@3]

And after the application of the Instant Database:

In the following section, provide the requested information regarding each child:		
Name of child	Address	Date of Birth
John Doe	123 Main Street	January 28, 1953
Paula Doe Smith	334 Oak Landing	July 7, 1952
Francis Doe	117 Cherry Orchard Land	August 18, 1962

- If you wish to repeat a selected row in a 'plain table' (that is, one that is not surrounded by any <<*Repeats*>> instructions, follow these steps:
 1. Place your cursor in the row you want to repeat. (In the below example, place cursor anywhere in the last row.)
 2. Click the Pathagoras features menu.
 3. Click the "Process" element and then the 'Repeat Rows' button.
 4. When prompted, type in the number of times you want the row to repeat. All rows at and below the cursor level will be duplicated and all [bracketed variables] will be incremented so they can be identified and answered separately when you run the the Instant Database.

In the following section, provide the requested information regarding each child:		
Name of child	Address	Date of Birth
[Child]	[Childaddress]	[Date of Birth]

Here is the result if "3" chosen:

In the following section, provide the requested information regarding each child:		
Name of child	Address	Date of Birth
[Child@1]	[Childaddress@1]	[Date of Birth@1]
[Child@2]	[Childaddress@2]	[Date of Birth@2]
[Child@3]	[Childaddress@3]	[Date of Birth@3]

Note: Inside rows cannot be duplicated as 'stand alone' units. All rows from the cursor position down will be duplicated. The limitation can easily be overcome by simply not having information you do not want repeated beneath the section of the table you want duplicated. If necessary, split the table.

18.7 'Return' the Repeat value

Returning the 'repeat' value. The above examples capture the number of requested repeats, but do not 'return' that value (i.e., '1', '2', 'three', etc.) to the editing screen. If you want to 'return' a value to your editing screen at the same time you make your selection, add the word 'return' in the administrative section of the 'Return' prompt (just after the Repeat command along with the 'style' of the returned number (numeral or spelled out.)

Examples (combining !groups! and 'returns'):

```
<<*Repeat(return,1)*!shareholders!>>
```

```
<<*Repeat(return,one)*!shareholders!*>>
```

- The "1" (or any other 1 or 2 digit number) after the second "*" tells Pathagoras to return the

Arabic numeral;

- The "one" (or any other number) tells Pathagoras to return a "spelled out" number.
- Whatever number you provide becomes the 'default' value when the 'how many' question is asked.

More examples:

```
<<*Repeat(return,one)*!shareholders!*>> individuals own stock in the corporation:
```

```
<<*Repeat*!shareholders!*Name: [ShareholderName] Shares Owned: [SharesOwned]
```

```
>>
```

The shareholders waive notification of the upcoming annual meeting as evidenced by their signatures below:

```
<<*Repeat*!shareholders!*_____
[ShareholderName]
```

```
>>
```

Here the result if the answer to "How many shareholders? is 3:

Three individuals own stock in the corporation:

Name: [ShareholderName@1] SharesOwned: [SharesOwned@1]

Name: [ShareholderName@2] SharesOwned: [SharesOwned@2]

Name: [ShareholderName@3] SharesOwned: [SharesOwned@3]

The shareholders waive notification of the upcoming annual meeting as evidenced by their signatures below:

[ShareholderName@1]

[ShareholderName@2]

[ShareholderName@3]

Another Example:

Children (the repeat's 'groupname' is "!cdn!")

Minor Children of the Parties

```
<<*Options*No Children/One Child/2+ Children*
```

Husband and Wife have no minor children from the marriage./


There is one minor child of the parties: [Child1 Name], born [Child1 Birth Date]./

There are <<*Repeat(return,two*!cdn!*>> minor children of the parties:
 <<*Repeat*!cdn![ChildName], born [ChildBirthDate]>>
 >>

Here is the result of the answer to "How many children? is '4':

Minor Children of the Parties

There are four minor children of the parties: [ChildName_1], born [ChildBirth Date_1] [ChildName_2], born [ChildBirth Date_2] [ChildName_3], born [ChildBirth Date_3] [ChildName_4], born [ChildBirth Date_4]

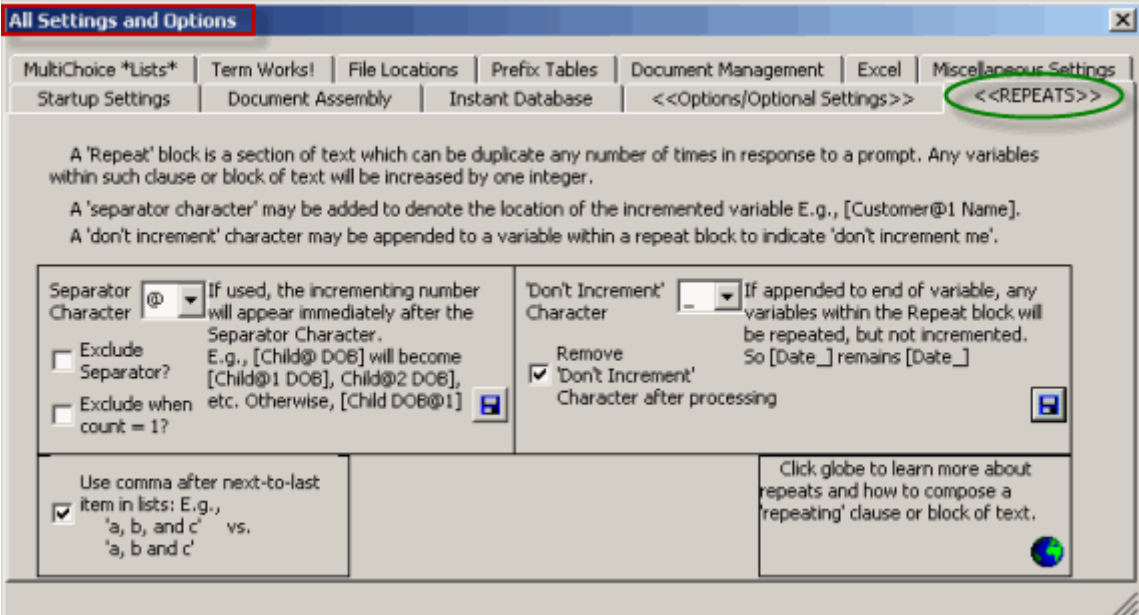
 If a value has previously been assigned to a !groupname! for a Repeat, you can return the spelled out value by simply enclosing the !groupname! within double angle brackets. E.g., from the above examples, <<!shareholders!>> and <<!cdn!>>.

18.8 Repeat Settings

The default 'increment marker' (the character that separates the base variable name from the incremented value) is "@".

Examples: The variable [Child] becomes [Child@1], [Child@2], etc.

You can choose a different marker, or choose not to have a marker at all. Make your choices at the 'Repeats' tab of the 'All Settings and Options' screen.



All Settings and Options

MultiChoice *Lists* | Term Works! | File Locations | Prefix Tables | Document Management | Excel | Miscellaneous Settings | Startup Settings | Document Assembly | Instant Database | <<Options/Optional Settings>> | <<REPEATS>>

A 'Repeat' block is a section of text which can be duplicate any number of times in response to a prompt. Any variables within such clause or block of text will be increased by one integer.

A 'separator character' may be added to denote the location of the incremented variable E.g., [Customer@1 Name].

A 'don't increment' character may be appended to a variable within a repeat block to indicate 'don't increment me'.

Separator Character: @ If used, the incrementing number will appear immediately after the Separator Character. E.g., [Child@1 DOB], [Child@2 DOB], etc. Otherwise, [Child DOB@1]

Exclude Separator? ☐ Exclude when count = 1? ☐

'Don't Increment' Character: _ If appended to end of variable, any variables within the Repeat block will be repeated, but not incremented. So [Date_] remains [Date_] Remove ☒ 'Don't Increment' Character after processing ☐

Use comma after next-to-last item in lists: E.g., 'a, b, and c' vs. 'a, b and c' ☒

Click globe to learn more about repeats and how to compose a 'repeating' clause or block of text.

If you want to exclude the separator altogether, check the Exclude Separator box at the left.

Sample result: [Child1], [Child2], etc.

If you want to change the separator itself, select the desired Separator Character from the dropdown list at the left.

Sample result: [Child_1], [Child_2], etc.

If you don't like the default placement of the at the end of the variable, 'tell' Pathagoras where to put the incrementing number by pre-placing the '@' sign in the 'repeat' text block itself. So, if a variable were written [Shareholder@ Address], it would increment [Shareholder@1 Address], [Shareholder@2 Address], etc.

Some variables within a Repeat block (such as [Date]) don't need to be incremented as the text block is being repeated. It will be the same value in all cases. You can tell Pathagoras not to increment a variable by inserting a 'Don't Increment' character as the last character of the variable name. The default 'Don't Increment' character is the underscore. The variable might look like this: [Date_]. (When the document is processed, the underscore will be removed, unless you uncheck the box that tell Pathagoras to do so.)

See Also:

['Repeat' Function \(general\)](#) 382

['Repeat' in Clause Selection Screen and DropDown Lists](#) 395

['Repeat' Alternatives](#) 395

18.9 'AskRepeat'

At or near the top of the document, you can insert a request for the user to provide a 'repeat' value for a designated groupname in this fashion.

`<<*AskRepeat*!groupname!*>>`. It is simply the 'repeat' keyword and the !groupname!. Nothing else. The entire prompt is 'administrative' which is why it closes with a '*'. Its sole purpose is to ask how many repeats of each !groupname! there are to be. The value is recorded, but nothing is returned to the screen.

You can augment the block with a question or prompt. E.g.:

`<<*AskRepeat*!numchildren!How many children were born of this marriage?*>>`"

Since the entire prompt is 'administrative', the entire prompt must close with a '*'.

See also:

[AskRepeat with Options](#) 392

[GroupNames](#) 151

[<<*AskIf*>> prompts](#) 506

18.10 Repeat with Options

When you ask for the number of repeats for a particular operator, Pathagoras will dutifully record the number of repeats and apply the result to other repeat block throughout the document. (5 children, it will create [Child1],[Child2], etc., blocks in the appropriate places throughout your document.

But what if what you also need is 'optional' text (not repeating text) based on the number of

something. For example, if there are no children of the marriage, choose an appropriate clause, if there is one child, choose a different clause and if there are 2 or more children, choose yet a different clause. This page addresses that.

Let's start with some sample text:

```
<<*AskRepeat*!Children!*>>
<<*AskRepeat*!Bene!*>>
(The above would appear at the top of the document. Pathagoras will always ask 'Ask'
blocks first.)

Last Will and Testament of
[Testator Name]

. . .

A. Family: <<*Options*!Children!*I have no children/I have one child whose name is/I
have <<*Repeat(return,two)*!children!*>> children and their names are <<*Repeat(and)
*!children!*[Child]>>>>.

. . .

<<*Options*!Children!*/
G. All trust assets shall be distributed to my child named in paragraph A if that child is
alive at the time of my death. If my child is not alive at the time of my death, but has
surviving descendants, the share that would have gone to such child shall be distributed to
such child's descendants, per stirpes./
G. All trust assets shall be distributed to each child named in paragraph A if that child is
alive at the time of my death. If a child of mine is not alive at the time of my death, but has
surviving descendants, the share that would have gone to such child shall be distributed to
such child's descendants, per stirpes.>>
<<*Options*!Children!*
H. All assets shall be distributed to <<*Repeat(and)*!Bene![AltBenef Name] residing at
[AltBenef Address]>>, if that beneficiary is alive at the time of my
death<<*Options*!Bene!*//,in equal shares>>. If a named beneficiary should not survive
me, such shares shall be distributed to such beneficiary's descendants, per stirpes./
H. If my child named in paragraph A, and no descendant of such child, shall survive me,
all assets shall be distributed to <<*Repeat(and)*!Bene![AltBenef Name] residing at
[AltBenef Address]>>, if that beneficiary is alive at the time of my
death<<*Options*!Bene!*//,in equal shares>>. If a named beneficiary should not survive
me, such shares shall be distributed to such beneficiary's descendants, per stirpes.>>
```

Note the following.

Paragraph A. contains 3 options that will be processed based on the number of children selected with the AskRepeat call. Notice that the text that will be returned is not based on the 'position' of the AskRepeat number. That is, if '1' is the response to the ask repeat, the Option in

position #1 is not what will be returned. When you study the example, you will see that that would make no sense. Read on:

EXCEPTION TO THE "POSITION" RULE Normally, !grouped! options are processed by 'position'. Whether you select the 1st member of the group or the 15th, the same 'position' of other members of the group is selected. However, when Repeats are !grouped! with an <<*Options*>> block, the selected option is processed based on an analysis of the actual number provided.

Zero: If the 'repeat' number is 0, the option text in position 1 of the <<*Options*>> block with the same !groupname! is processed.

One: If the 'repeat' number is 1, the option text in position 2 of the <<*Options*>> block with the same !groupname! is processed.

Two or more: If the 'repeat' number is 2 or higher, the option text in position 3 of the <<*Options*>> block with the same !groupname! is processed.

There is a real logic behind this operation and it deals with noun/verb pairings:

If there are 'none' of something, the plural is used to describe this 'nothingness.'

If there is 'one' of something, the singular is used.


If there are '2 or more', the plural is used.


For example:


"There *are* no children. There *is* one child. There *are* 3 children."

Further, the actual text (if any text is provided at all) may vary greatly based on the number of children, shareholders, quantity ordered, etc.

So, when using the same !groupname! with a <<*Repeat*>> block and an <<*Options*>> block, simply set out the appropriate choices for the '0', '1' and '2+', possibilities, separating each choice with slashes.

 If there is no appropriate text to display for one of the choices, you will still need the slash. It then just serves as a holding spot. See just above Paragraph G ('0' children requires no text), and see in Paragraph H near the !bene! group ((there is no 'equal shares' possibility for '0' or '1' beneficiary).


"But what if I need separate paragraphs based on all of possibilities reflected in the AskRepeat answer?" Sorry, but we simply cannot help. We had to make a decision as to how best to process the 'repeat' answer and could not conceive of the above situation where a user would need 10 'Options' choices if '10' was the answer to the AskRepeat. (If you simply need to return the number '10' within the Options text, that is easy enough to set up. See [Returning the Repeat value](#))

 **NOTE:** By design, Options blocks will always be processed before Repeat Blocks. This is so even when the Option block is nested within the Repeat block. (In any program, there must always be a processing order or precedence. Here, with the exception described immediately below, the order is 'Options and Optional' blocks first, and then Repeats.) If you want the document's Repeat blocks to process first, simply present an 'AskRepeat' prompt at the top of the document. This is shown in the example.

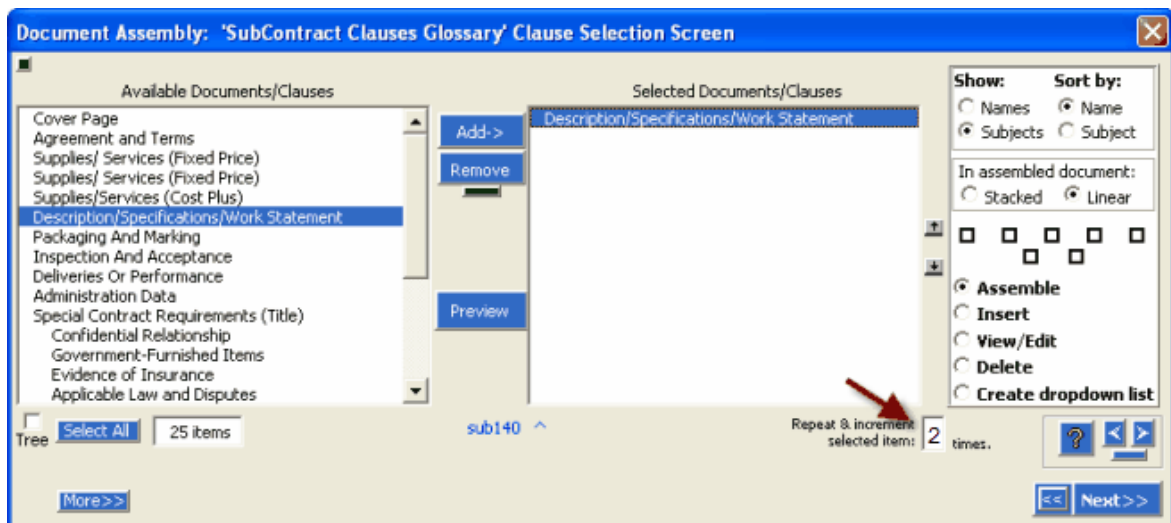
If you see your Repeat blocks automatically processing when it bears the same groupname as an Options block, that is because of the above rule. Here's what's happening. When the values

for the Options block are presented, and you select one, the value for the !groupname! is immediately set. Then, when the Repeat block is encountered, its value is 'known' to Pathagoras and the Repeat block is repeated the 'appropriate' number of times, based on the Options position selected.

18.11 Repeats Elsewhere




 There are two additional 'repeat' functions which Pathagoras makes available:

1. The **Clause Selection Screen** provides a 'Repeats' box that displays when a single non-set element has been selected and moved to the right panel. When completed, Pathagoras will insert the selected clause the designated number of times.



The Repeat Function is available in the Clause Selection Screen when a single element has been selected.

See Also:

['Repeat' Function](#)  382
['Repeat' Settings](#)  391
['Repeat' Alternatives](#)  395

18.12 Repeat Alternatives

There are other methods available in Pathagoras which mimic the 'Repeats.' They offer more controlled results in terms of 'variable naming.' However, this may be more information that anyone needs. Read on only if the above 'Repeats' functions do not serve your needs.

The easiest and most straightforward solution is to prepare substantially identical clauses to the one you want repeated. Each clause differs from the others in that they reflect 2, 3 or 4, etc., of the element sought to be repeated. This solution requires no coding on your part. It just requires that you to save additional clauses to reflect the desired ends.

For example, in a Will, you might have the following alternative clauses reflecting from zero to four children:

[wil120a:](#)

There were no children born of this marriage.

[wil120b:](#)

There was one child born of our marriage, namely [Child1Name] born [Child1DOB].

[wil120c:](#)

There were two children born of our marriage, namely [Child1Name] born [Child1DOB] and [Child2Name] born [Child2DOB].

[wil120d:](#)

There were three children born of our marriage, namely [Child1Name] born [Child1DOB], [Child2Name] born [Child2DOB], and [Child3Name] born [Child3DOB].

[wil120e:](#)

There were four children born of our marriage, namely [Child1Name] born [Child1DOB], [Child2Name] born [Child2DOB], [Child3Name] born [Child3DOB] and [Child4Name] born [Child4DOB].

When you are assembling a clause, simply select the appropriate 'family size' clause from the list of clause options. This technique is straightforward and easy to implement.

=====

The same result can be achieved by using a combination of two powerful Pathagoras functions, namely [MultiChoice *Lists*](#)^[458] and [{Simplified Options}](#)^[164]. A MultiChoice *Lists* is just a string of alternative text, separated by slashes ("/") and stored in the MultiChoice *List* table. (Creating and editing these *Lists* is very simple.)

Instead of saving the permutations as separate documents (in the fashion described above), you can save them as part of the MultiChoice *List* collection:

Step 1: Create a MultiChoice *List* called, for example, "children." Include the following variables as its value:


There were no children born of this marriage/There was one child born of this marriage, namely [Child1Name], born [Child1DOB]/There were two children born of this marriage, namely [Child1Name], born [Child1DOB] and [Child2Name], born [Child2DOB]/There were three children born of this marriage, namely [Child1Name], born [Child1DOB], [Child2Name], born [Child2DOB], and [Child3Name], born [Child3DOB]/There were four children born of this marriage, namely [Child1Name] born [Child1DOB], [Child2Name] born [Child2DOB], [Child3Name], born [Child3DOB] and [Child4Name], born [Child4DOB]/There were five children born of this marriage, namely [Child1Name], born [Child1DOB], [Child2Name] born [Child2DOB], [Child3Name], born [Child3DOB], [Child4Name], born [Child4DOB] and [Child5Name], born [Child5DOB]

Step 2: Insert the following text in the source document:


There were [number of children] children born of our marriage, namely { *children* }.


When the document is assembled, Pathagoras will pause at the Options text { *children* } and present the options contained in the *children* list. Select the appropriate one.

This technique is further illustrated near the bottom of the [MultiChoice *Lists*](#) ⁴⁵⁸ page on this site.

 The same result can also be achieved by using Pathagoras regular <<*Options*>> formula with questions assigned, and setting up a appropriate MultiChoice *List* for 'children' as described above. The MultiChoice *List* for children is identical to that which is described immediately above, but the text introducing the <<*Options*>> block which introduces the questions is a little more complex. Here is an example. Try this:

```
<<*Options*One Child/Two Children/Three Children/Four Children/Five
Children**children*>>
```

 A similar result can also be achieved by using Clause Sets. Clause Sets are virtual documents that contain pointers to clauses, the entire collection of which represents a complete document. You can design individual clauses and then a Clause Set to represent a complete document for (following the above example) 'no children', 'one child', and 'multiple children'.

 Here is one more example to illustrate 'repeats.' It is similar to the first example above. Each clause is created following the same format, with the name of each document reflecting the number of repeats desired. So, let's say you want to prepare for cases where you have multiple buyers and sellers to a real estate deal.

Save documents named, e.g., 'Buyers1.doc' with the content:

[Buyer Name] of [Buyer Address], party of the second part.

a second document called 'Buyers2.doc' with content:

[Buyer1 Name] of [Buyer1 Address] and [Buyer2 Name] of [Buyer2 Address], parties of the second part.

...

a fifth document called 'Buyers5.doc' with content:

[Buyer1 Name] of [Buyer1 Address], [Buyer2 Name] of [Buyer2 Address], [Buyer3 Name] of [Buyer3 Address], [Buyer4 Name] of [Buyer4 Address], and [Buyer5 Name] of [Buyer5 Address], parties of the second part.

. . . etc. (A similar set of terms could be created for multiple sellers.)

The source clause might read:

This Contract made by and between { Sellers1/Sellers2/Sellers3/Sellers4/Sellers5 } and { Buyers1/Buyers2/Buyers3/Buyers4/Buyers5 }

When assembling the new document using the source clause, Pathagoras will display the { options } text in a selectable list. Select the appropriate item. Pathagoras will find the appropriate clause and insert it where appropriate.

18.13 Repeat Restrictions

Nesting:

- A <<*Repeat*>> block can be nested within itself but only to one level.

Structure:

- A 'groupname' for a Repeat block must be enclosed within "!"s.
 - This structure is legal: <<*Repeat*!Shareholders!* . . . >>
 - This structure is not legal: <<*Repeat*Shareholders* . . . >>
- All text following the closing of the 'administrative text' will be included. So if you wanted to have a group name called '!Shareholders!' but also want the word "Shareholder:" to appear in the repeated text block, just add it after the groupname. Here is an example:

<<*Repeat*!Shareholders!*Shareholder:

[Shareholder]	[SharesOwned]
[ShareholderAddress]	

>>

See [Administrative text](#) 162.

18.14 Testing Repeat Blocks

You can test the structure, integrity and action of any <<*Repeat*>> block by either 'assembling' a document or by running 'Process' against the text. [Click here to read more about Testing your documents.](#) 160

18.15 Repeat Example

Lawyers typically have special requirements for the preparation of 'pleadings' and other court-filed documents. Here are some examples of how Repeat blocks can be used to prepare a 'style' of the case.

More samples and examples of captions and signature blocks can be found [here](#)⁶⁷⁹ as well.

To keep things aligned, you may wish to use a 3 column table, the left side of which contains the parties' names, the center (very narrow) column of which contains the 'squiggles' and the right column of which contains the case info, etc.

Examples are provided below. You should be able to highlight, copy and paste any of these examples into your document. However, some margin and table settings could be lost in the translation. If the copy/paste is not accurate, you can download the identical examples from this URL. [Repeat \(Case Captions\).doc](#)

(Remember, the colors are for illustrative purposes. 'Painted' text is not required.)

```
<<*AskRepeat*!Plaintiffs!*>>
<<*AskRepeat*!Defendants!*>>

<<*Repeat(and)
*!Plaintiffs!*[PLAINTIFF NAME]
>>
, Plaintiff{ !Plaintiffs!/OR/ORs }
vs.
Case Number: [Case No.]

<<*Repeat(and)
*!Defendants!*[DEFENDANT NAME]
>>
, Defendant{ !Defendants!/OR/ORs }
```

MOTION FOR RELIEF

Now come your <<*Options*!plaintiffs!*Plaintiff/Plaintiff/Plaintiffs>>, <<*Repeat*!plaintiffs!*[Plaintiff]>> and state the following:

1. ...

```
=====
<<*AskRepeat*!P!*>>
<<*AskRepeat*!D!*>>
```

The following is the identical setup as the above except we have shortened the groupnames to a single character. (The shorter the groupname, the better the appearance, but the groupname must be 'clear' enough to the actual operator.

```

<<*Repeat(and)*!P!*[PLAINTIFF
NAME]
>>
        , Plaintiff{ !P!/OR/ORs }
        vs.
        ) Case Number: [Case No.]
<<*Repeat(and)*!D!*[DEFENDANT
NAME]
>>
        , Defendant{ !D!/OR/ORs }

```

MOTION FOR RELIEF

Now come your <<*Options*!p!*Plaintiff/Plaintiff/Plaintiffs>>,
 <<*Repeat*!p!*[Plaintiff]>> and state the following:

1. . . .

=====

Here is an example with the split caption frequently used in Bankruptcy courts

<<*AskRepeat*!Debtors!*>>

IN THE UNITED STATES BANKRUPTCY COURT FOR THE [EASTERN/WESTERN/NORTHERN/SOUTHERN] DISTRICT OF [STATE] [DIVISION NAME] DIVISION

```

<<*Repeat(and)
*!Debtors!*[DEBTOR NAME]
>>
        , Debtors{ !Debtors!/OR/ORs }

```

```

<<*Repeat(and)
*!Debtors!*[DEBTOR NAME]
>>

```

CHAPTER 13

CASE NO. [Case Number]

, Movant{ !Debtors!/OR/ORs }.

vs.

```

[SECOND MORTGAGE HOLDER]
        , Defendant

```

CONTESTED
 PROCEEDING

MOTION TO AVOID LIEN

Now come your Debtor{ !Debtors!/OR/ORs }, pursuant to 11 U.S.C. Section 506(a) and (d) and Fed.R.Bankr.P. 3012,

=====

18.16 Repeat (a 'one page' lesson)

Frequently you will have a form document where the number of actors (plaintiffs, defendants, shareholders, beneficiaries, buyers and sellers, etc.) varies from from client to client.

Solution 1: It is possible to 'hard code' all the possible variations into the form using Options (e.g., "no children/one child/two children/three children" and so on). If the number of options is fixed and small, that frequently is the best solution.

Solution 2: Where the number of possible options is large or infinite, Solution 1 is not terribly efficient. You should use Pathagoras "Repeat" function in those cases. The "Repeat" function is simple to add, but very powerful in application.

'Repeat' tells Pathagoras to ask "how many actors" to account for. When answered during a document assembly session, it causes Pathagoras to insert a separate, distinct variable for each actor. ('Distinct' means not only is the variable repeated the proper number of times, but an incrementing number is inserted into the variable name so that it is slightly different from the others in the same set.)

For example, if your basic form includes a variable called [Shareholder Name] and you are drafting a document with 3 shareholders, you might insert the Repeat command into the source document like this:

<<*Repeat*[Shareholder Name]>>

(Note: the colors are for illustration and emphasis only.
They are not required for the repeat function to properly function.)

At document assembly time, Pathagoras will locate the repeat block, pause and ask for the number of Shareholders. When the end user supplies an answer (let's say it is '3'), Pathagoras will delete the original variable and replace it with designated number of distinct variables:

[Shareholder Name1] [Shareholder Name2] [Shareholder Name3].

When you are ready to personalize the document using Instant Database, Pathagoras will see the new variables, display them onto the Instant Database screen along with all other document variables for completion.

Can more than one piece of information appear within a Repeat block? Yes. Continuing with the above example, let's say you also want to include addresses and the number of shares for each shareholder. The setup is identical. Just add the more variables as desired:

<<*Repeat*[Shareholder Name] [Shareholder address] [number of shares]

>>

(note that the closing '>>' is on a separate line.

That signals Pathagoras to put each variable set on a separate line.)

If the end user tells Pathagoras '3' during document assembly time, Pathagoras will return these lines of text:

```
[Shareholder name1] [Shareholder address1] [number of shares1]

[Shareholder name2] [Shareholder address2] [number of shares2]

[Shareholder name3] [Shareholder address3] [number of shares3]
```

Can I repeat an entire document?

Absolutely. Suppose you want to send your shareholder agreement to each shareholder. You can draft a cover letter which will also have incrementing variables so that one letter will be drafted for each shareholder. Your letter might look like this:

```
<<*Repeat*
```

```
[Shareholder Name],
[Shareholder Address]
```

```
Dear [Shareholder Salutation],
```

```
Enclosed is a draft of the shareholder agreement. Please call me with your
comments and questions.
```

```
Sincerely,
```

```
Arnie Attorney
```

```
>>
```

If the end user tells Pathagoras to generate '3' letters, the result will be a single document contained three distinct letters:

```
[Shareholder Name1],
[Shareholder Address1]
```

```
Dear [Shareholder Salutation1],
```

```
Enclosed is a draft of the shareholder agreement. Please call me with your
comments and questions.
```

```
Sincerely,
```

```
Arnie Attorney.
```

```
[Shareholder Name2],
[Shareholder Address2]
```

```
Dear [Shareholder Salutation2],
```

```
Enclosed is a draft of the shareholder agreement. Please call me with your
comments and questions.
```

Sincerely,

Arnie Attorney

.
[Shareholder Name3],
[Shareholder Address3]

Dear [Shareholder Salutation3],

Enclosed is a draft of the shareholder agreement. Please call me with your comments and questions.

Sincerely,

Arnie Attorney

(If you had an End Page or Section break within the Repeat block, they would be duplicated along with the text.)

But wait, there's more!!! (just like Ronco)

What if I want to generate multiple documents, but there are certain variables that I do not want to be incremented. Can I do that?

Of course you can. You simply need to tell Pathagoras which variables do not increment. Your letter might look like this:

```
<<*Repeat*
[Date of Letter_]

[Shareholder Name],
[Shareholder Address]

Dear [Shareholder Salutation],

Enclosed is a draft of the shareholder agreement. You can see that the
corporation will issue [total number of shares issued_]. You will receive
[number of shares] of those shares. Please call me with your comments and
questions.

Sincerely,

Arnie Attorney.
>>
```

Did you notice the extra _ at the end of [Date of Letter_] and [total number of shares issued_]? That underscore tells Pathagoras not to increment the variable. (The date of the letter and the total number of share issued will always be the same from document to document. By default, Pathagoras will remove the ending '_' during processing.)

In the above examples the incrementing number always appeared at the end.

Can I tell Pathagoras to place the incrementing number somewhere else?

Yes. Just put the symbol “@ “ at the spot in the variable where you want the number to be placed. Eg., [Child@ Name] will become [Child@1 Name], [Child@2Name], [Child@3 Name], etc.

The above examples simply repeated the variable, without any connecting commas and the word ‘and’ before the final variable.. Can I add list ‘punctuation’ to the repeat block?

Yes. Just add the appropriate ‘list signal’ to the Repeat command. Examples:

```
<<*Repeat (and)*[Child]>>
```

becomes (with the answer 3)

```
[Child1], [Child2] and [Child3]
```

```
<<*Repeat (and;)*[Child@ Name], born [Child@ DOB]>>
```

becomes (with the answer 3)

```
[Child@1 Name], born [Child@1 DOB]; [Child@2 Name], born [Child@2 DOB] and  
[Child@3 Name], born [Child@3 DOB]
```

Can I tell Pathagoras to use the same number of repeats at disparate locations in the document?

Yes. Just add a group name to the Repeat block, just after the repeat command. (A groupname is any text between two exclamation marks.) Close the extended ‘administrative text’ section with a third asterisk. The groupname E.g.,

```
<<*Repeat*!shareholders!*>>
```

The following individuals own stock in the corporation:

Name

Shares Owned

```
<<*Repeat*!shareholders!* [ShareholderName], [SharesOwned]
```

```
>>
```

(. . .)

The shareholders waive notification of the upcoming annual meeting as evidenced by their signatures below:

```
<<*Repeat*!shareholders!* _____  
[ShareholderName]
```

```
>>
```

The Pathagoras System

Clause Sets

Part



XIX

19 Clause Sets

A 'clause-set' is a 'potential' complete document. It is (typically) a simple collection of 'pointers' to previously saved 'building block' text.

When a clause-set is called upon, Pathagoras locates the building blocks indicated by the pointers and inserts that text into a new, or into the current, document.

Clause-sets are ideal for standard documents which involve the selection of identical (or close to identical) clauses each time. For example, a simple 'I Love You Will' or a standard 'Real Estate Purchase Contract' typically contains the same pattern of clauses.

Whenever you perceive that it is wasting time to repeatedly check off the identical clauses whenever such a document is needed, that's the time to create a clause-set.

But don't stop there. Clause-sets are very helpful for creating those documents that you rarely need. If you create a 'rare' document and believe it may be difficult for you or another to recall which clauses constitute a 'basic' form of that genre of document, create a clause-set for that one as well.

Benefits of Clause-sets:

- Simple to very complex documents can be pre-created.
- Since the building blocks are stored and updated independently from a clause-set, the 'freshest version' of the building blocks will always be called into the final document.
- Clause-sets are easily editable. A clause-set is stored as a *simple Word document*. As such, it can be recalled, edited and re-edited.
- Clause-set can be used within other clause-sets. Very complex document can be constructed through their use.
- You can create a clause-set for each possible contingency. You can name your various clause-sets in such a way as to tell the end user which one to select for a particular circumstance. This can mimic the often complex process that 'interview driven' programs such as HotDocs(r) use to select the proper clauses for assembly.
- You can use clause-sets in conjunction with [<<*Options*>> blocks](#)¹⁴⁶ to mimic (without the programming complexities) the often challenging process that 'interview driven' programs such as HotDocs(r) use to assign the proper clauses for assembly. See also [Calls to Other Documents](#)¹⁰⁴ for illustrations of how to assign clauses (including clause-sets) to <<*Options*>> blocks.
- While not precisely the same, implementation of above can also bring powerful "If . . . / Then . . ." logic, including cascading answer groups to your documents. See a further discussion of this at [this link](#)⁴⁹².

The 'technical' side of Clause Sets:

When you create a Clause Set, Pathagoras generates a simple text file stored in the book that contains the names of the clauses/files that you designated as part of the set. Pathagoras simply reads down the list and pours each new document into the document being assembled.

Caveat: If a template was associated with the book from which the clauses set was drawn, Pathagoras first lays down that template as the 'first' document and then pours the clauses into there. If

no template was assigned, Pathagoras uses the first document as the template for the designated clauses.

If the margins or styles in the second, third, etc., clauses are inconsistent with the margins and styles of the template (or the first) document, you should consider the steps [outlined at this link](#)⁶⁷.

Pathagoras presumes that all clauses are from the same folder. Therefore, full paths to the desired clauses are not required for the set. But if a desired document from the Set is not in that folder, you can manually edit the clause set and insert a full-path reference to a document.

To edit the Clause Set, just navigate to it in Windows or Word. Open it up as a text document. Add a new row to the table and manually type, in the left column, the full name of the document being added, including path if not in the same folder as the other clauses. (You should add a subject in the right side as well.)

19.1 Creating Sets from Clause Selection Screen

To create a clause set 'automatically', run through a typical document assembly routine:

1. Select the desired book of clauses from the Libraries & Books screen and press **Next>>** to display the Clause Selection screen;
2. Select the clauses that you want the clause-set to comprise. Figure 1 displays a sample of what your screen may look like.

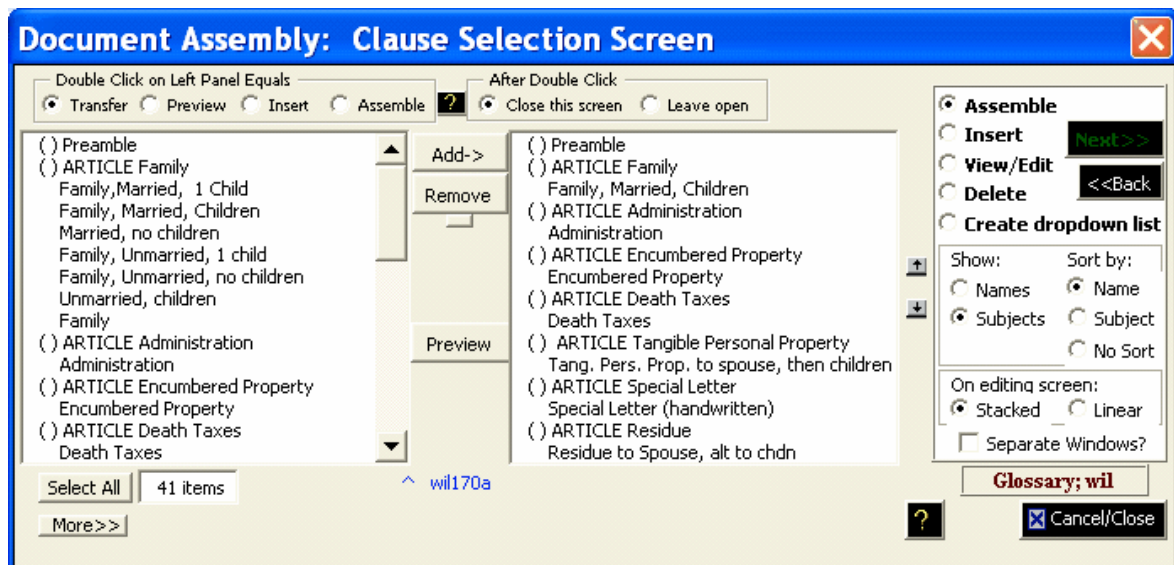


Figure 1. The Clause Selection Screen (default view). Clauses in the right panel are ready for assembly.

3. Click the More>> button in the lower left side of the screen. The screen expands and displays several additional controls. Click “Create clause-set.” (See Figure 2.)

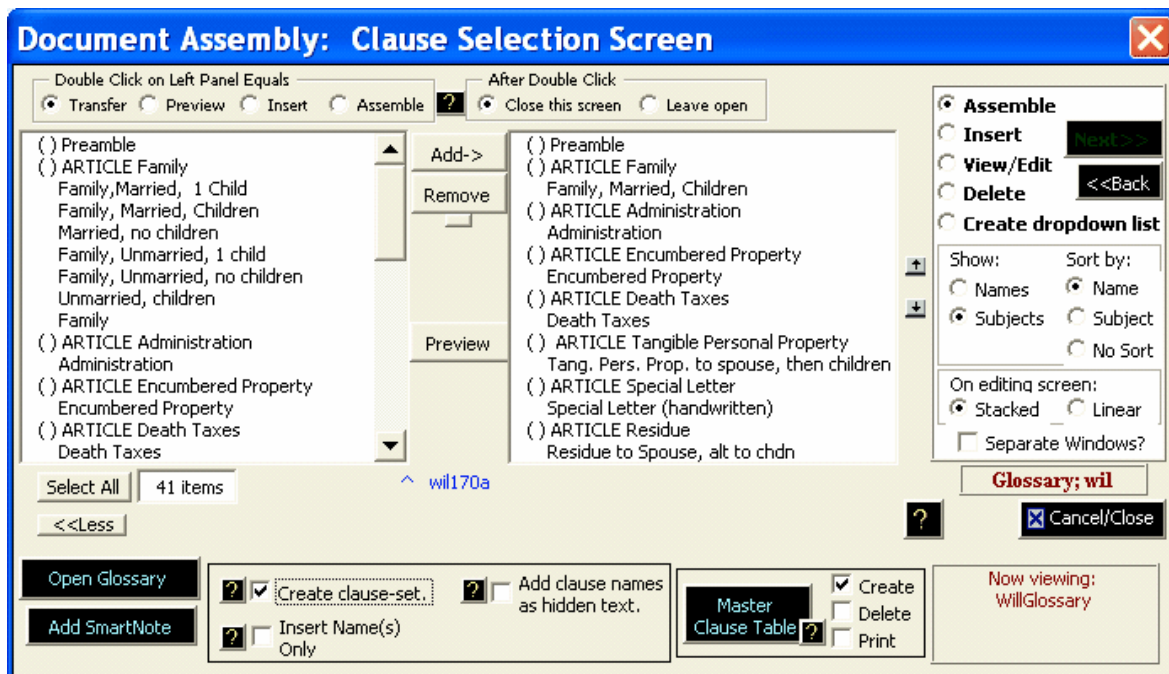


Figure 2. The Clause Selection Screen ("More" view)

4. Press the **Next>>** button in the upper right corner of the screen. The selected clause *names* are inserted into a table in a new document. (If clauses are drawn from a glossary, they are saved in the glossary.)
5. Save the clause-set as a new document or glossary term. Pathagoras takes you by the hand in naming and filing the set. More on this below.

Example:

Here is what a clause-set table might look like after it is saved as a source document (or saved into the source glossary). This example is 5 clauses 'long'. Clause-sets can be of any length, including hundreds of clauses long.

ClauseSetTable(C:\office forms\contract clauses)	
con100	Preamble
con150	Recitals
con130	Price and Performance
con140	Conditions
con190	Signatures

Here is how to 'read' a clause-set:

- The row of text above the table tell Pathagoras that this is a Clause-set 'table' and (in parenthesis) the source of the clauses. In the example, the clauses all come from the "C:\office forms\contract clauses" folder.

- A two column table follows. Only the left column is 'important.' It contains the clause names. The right column displays the subject given to the term (if any). It is helpful for the user to have the subject associated with the actual clause name, but the subject is not needed by Pathagoras. Because of this fact, you can freely change the content of the right column to make it even more meaningful to you or others who might view the table.

Saving the Clause-set

Once you have 'perfected' your clause-set, you should save it to a proper location. If you drew the clauses from a folder of documents, save the clause set as a new document in the same folder in which the source clauses exist. If the terms came from entries in a glossary, save the clause back into that same glossary from which the clauses were drawn. Pathagoras actually will take you automatically through this process.

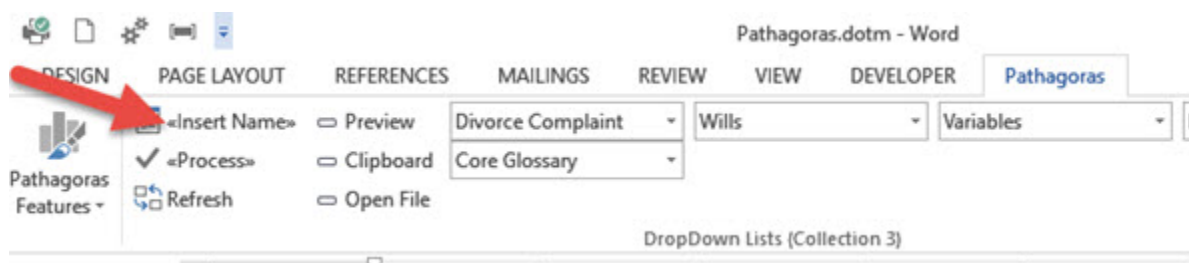
For most effective use, you should name the clause-set with the prefix 'set_' at the beginning of its name.

Editing the Clause-set

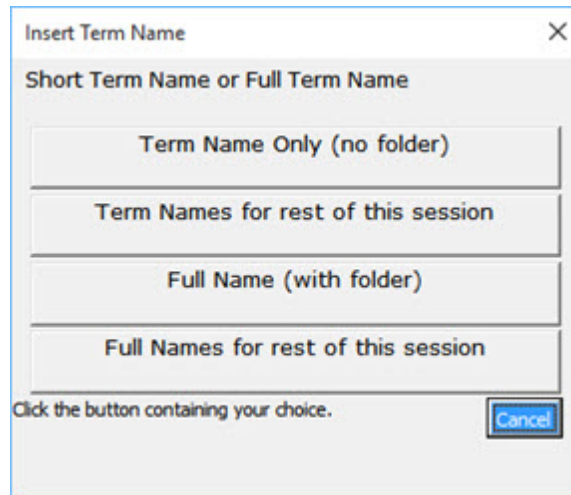
If you want to edit the clause-set (whether before or after you save it, feel free to do so. You can add new clauses to the clause-set table by typing in the term name following the pattern of the table you see displayed in the new source document. See the below section on [Editing Clause-sets](#) ⁴¹³.

19.2 Create Set from DropDown List

If you have previously assigned one of your DropDown Lists to a folder of text snippets from which you wish to create a Clause Set, you will find it incredibly easy to do so. Toggle the button in the upper left quadrant of the DropDown List section until it reads <<Insert Name>>. Then select one item from any of your DropDown Lists to begin building your Clause Set.




When you click your first item from your DropDown List, you will see this menu. Make a choice as to how you want your selection to be presented (short name or fully qualified (full address) name. Continue making choices from the DropDown List. (Note: you can mix and match! Nothing limits you to a single list.)



So, by selecting the appropriate terms, you have quickly built a Clause Set. See below:



Once you have the clauses you desire, save your Clause Set under an appropriate name using regular Word tools, but don't forget: you can save your Set into the folder represented by one of your DropDown Lists by clicking the 'Add text to folder' at the foot of the List. Then when you recall the document from the List, it will automatically process.

 You are not required to save the on-screen list as a Clause Set. Perhaps you wanted to just drop in clauses and process your list without saving it. Not a problem. You can process what is on the screen at any time. Just press <Alt-P>.

19.3 Creating Clause Sets Manually

A clause-set can be *manually* created. That's because, as we like to reiterate, Pathagoras is a plain-text document assembly tool. It allows you to type from your keyboard and knows how to interpret what you have typed.

So, you can simply type onto any editing screen the desired clauses in the order in which you wish for them to appear in the final document. You need only enclose each clause reference between "<<" and ">>" braces (no quotes).

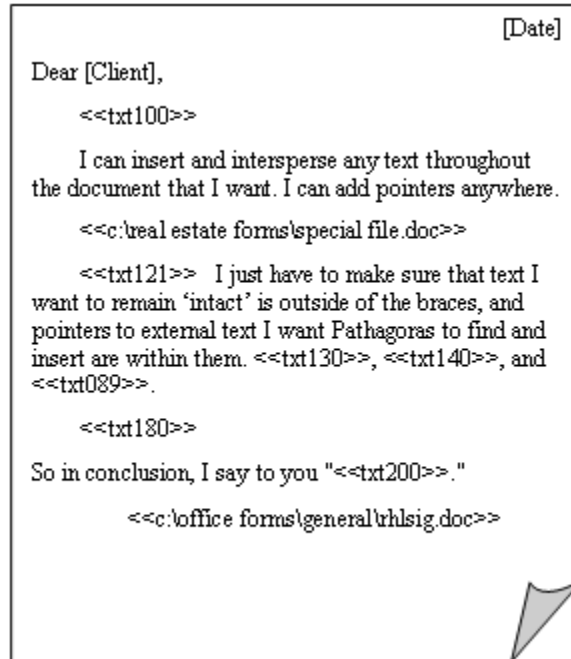
Note: The "<<" and ">>" marks are not 'keyboard codes' that require special keystrokes to insert. They are the 'lesser than' and 'greater than' symbols. Find them above the 'comma' and the 'period' on your keyboard.

The following text illustrates and more fully develops the process of manually creating clause-sets.

By using "<<" and ">>" markers to create a clause-set, your final product can comprise nothing other than clause references. Check this out:

```
<<txt100>>
<<txt110>>
<<txt124>>
<<txt124a>>
<<txt130>>
<<txt150>> <<txt155sp>>
<<txt200>>
<<txt223_simple>>
<<txt300_rhlsig>>
```

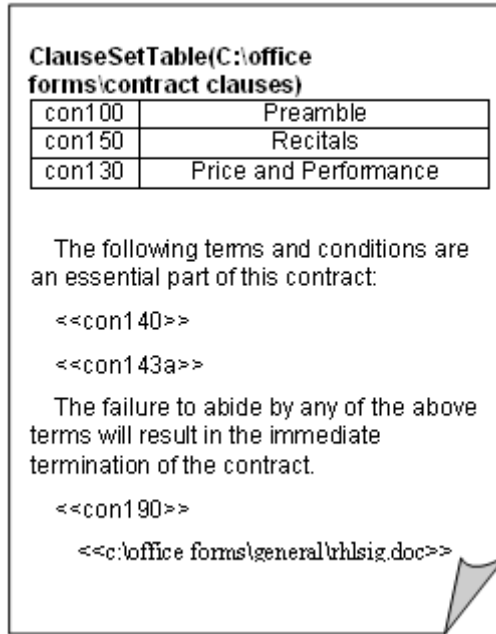
Your clause set can also commingle normal text and <<bracketed text>>. In the below example, the letter contains regular text mixed with <<document calls>>. A few variables are thrown in to show how you can create a complete 'package'. Just beneath this example, we dissect its content and composition to help you in creating your own clause-sets.



Note in the above letter that some *<<document calls>>* are simple prefixed/suffixed references, and others are 'fully qualified' references. If the names in the document source follow the prefix/suffix naming style, the short-name is all that is needed. Pathagoras will always follow the [Order of Search](#)⁹⁵ rules to find documents you have called.

Pathagoras is exceptionally flexible regarding how it uses and processes clause-sets. You can mix and match Clause-Set Tables (those created using the 'automatic method' and *<<pointers in braces>>*). Those in tables are processed as a unit. But you can type in freehand a clause name within the table (just add another row) or remove an item from the clause table (delete a row). You can remove it from the table and manually place it into the body of your new clause-set. Just make sure the manually inserted term appears between braces (e.g., *<<con130>>*). You can change the order of the clauses. You can add anything else you want into the document (pictures, boxes, lines, etc.). Pathagoras will process both the table and the *<<newterm>>* as it (and in the order that it) encounters them.

Here is a re-write of the 'contract' shown on the 'Creating Clause Sets Automatically' screen with the 'mixed' elements discussed in the immediately preceding paragraph.



See also: [Creating Clause Sets From Clause Selection Screen](#) ⁴⁰⁷

[Creating Clause Sets From DropDown Lists](#) ⁴⁰⁹

[Prefix/Suffix Naming Style](#) ⁴²⁰

[Double Angle Bracket Functions](#) ¹⁰⁴

19.4 Editing Clause Sets

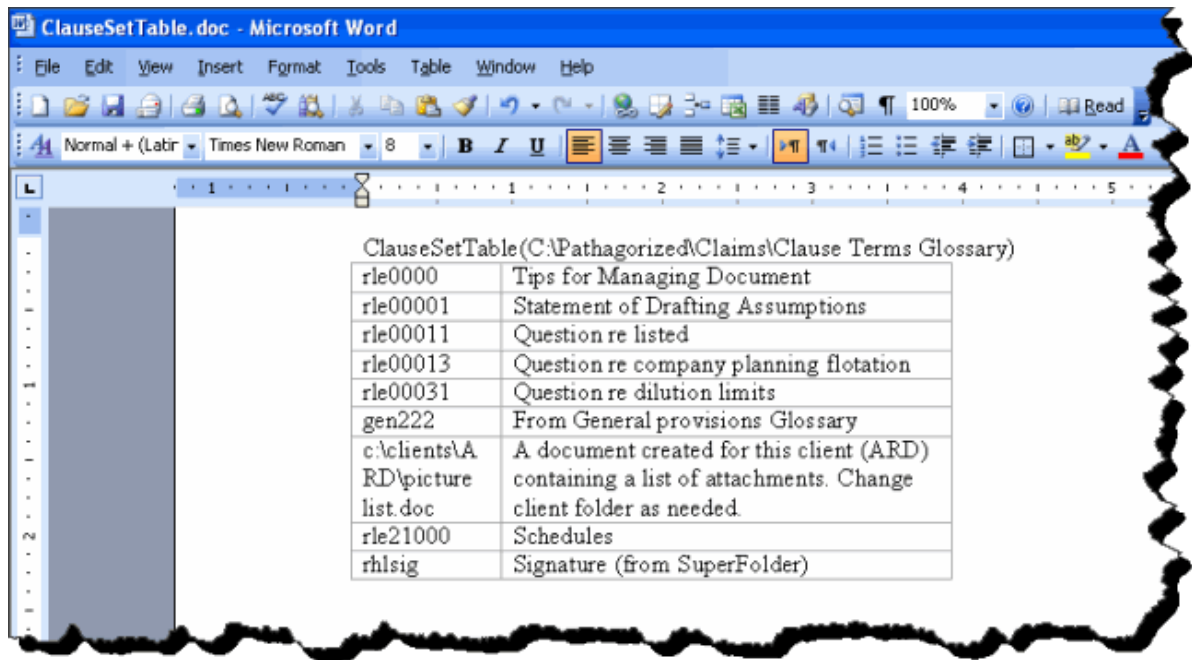
As indicated in previous sections, a Clause Set is a Word document (or a clause within a Word document). As such, it can be edited just like any other Word document.

To edit a clause set, simply navigate to its location. If it is in a folder of clauses, you can manually navigate to it or use the initial Document Assembly screen to select the appropriate book and then open the folder which contains the various clauses. Select the clause-set and edit away. If the book is a glossary, open the glossary, locate the clause-set and edit as appropriate. Remember to save your work.

While editing, you can manually type additional entries. You can add entries at the beginning, in the middle or at the end of the current display. To add at the end of the table, just place your cursor in the last row and press 'Tab'. That will start a new row. To add anywhere else, use Word's 'tables' controls to insert additional rows.

You can mix and match entries. While the initial 'build' of the Clause Set table will likely be from a single folder or glossary, you can mix up the table to you heart's content, drawing text from other folders or other glossaries. Just remember that Pathagoras' standard [search order rules](#) ⁹⁵ apply. If Pathagoras could find the document in any other setting, Pathagoras can similarly find it if contained in a clause set table.

Here is an example of a fairly complex clause set. Pathagoras is forced to look in two different glossaries, in the SmartPath and is required to use the prefix-table to locate the various terms.



19.5 Document Assembly with Clause Sets

Build a document using a clause-set via the Clause Selection Screen:

1. Click the Document Assembly icon.
2. Select the book that contains the clause-set.
3. Select the 'Show Clause Sets Only' option from the 'Display Options' box that appears.
4. Click <Next> to display the Clause Selection screen.
5. Choose the appropriate set from the Clause Selection screen.
6. Optional: If you want to display the contents of the clause set in order to remove specific clauses which are not appropriate for this particular client or customer, click the <Expand> button that appears between the two panels. All clauses comprising the clause set will be shown. Remove non-desired clauses by selecting the clause from the left panel and click the <Remove> button.
7. Select <Assemble> or <Insert> and then click <Next>.

Mouseless Assembly using a Clause Set:

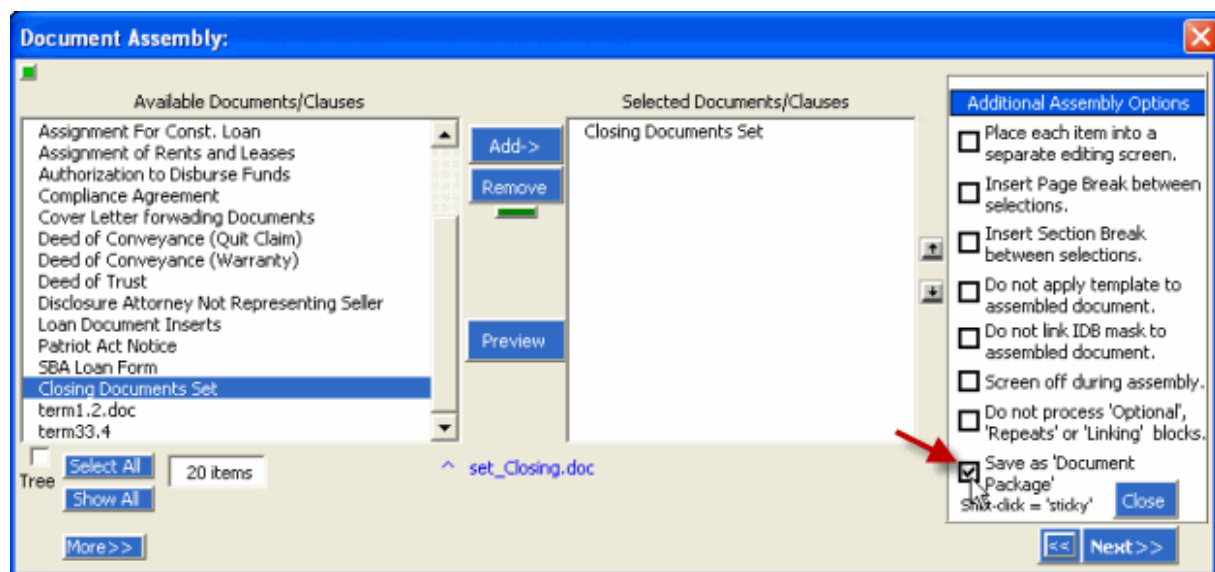
If the clause-set has been built on clauses which follow the 'prefix/suffix naming convention,' and if you know the clause-sets name, you can avoid the Clause Selection and all intermediate screens. Simply type the name on any blank line on your editing screen and press <Alt-G>. Pathagoras will find it for you directly. (You can even omit typing the

“set_” prefix.) For this reason, it is a good idea to keep a checklist of set_clauses handy. See [Checklist](#)⁴²².

Document "Packages"

Please review this section of the manual discussing [Document Packages](#)⁵⁸. It is essential that you understand what a Document Package is before proceeding with the following.

If the individual elements in a Clause Set are themselves complete documents, you can take advantage of the power of the 'clause set' function. Simply pre-create the document package as a Clause Set. Then, when you are ready to create the documents, display the appropriate book in the Clause Selection Screen, select the Clause Set from the left panel and 'add' it to the right. Then, click the 'Assembly Options' button and click the "Save as Clause Package" box at the right.)



When you tell Pathagoras that you want to save the documents as a Package, it saves each document listed in the right panel as a separate document in a folder (typically a client's folder) that you are asked to navigate to. Pathagoras then opens up each document in individual Word screens. *If a clause set was selected, Pathagoras first 'expands' the Clause Set to show its individual components in the right panel of the Clause Selection Screen.* You would then edit each document. When finished editing, you would (or course) save it. The beauty of a Clause Package is that each document you are now editing is *already saved in the proper client folder*. It also carries the name of the original document, which you can rename if desired. Document that may have had different margins or other formatting or styles from other documents in the package still retain those attributes. And finally, when you run the Instant Database to replace variables, Pathagoras will replace those variables *across all open documents*. (No need to run the IDB against each individual document.)

19.6 Expanding Clause Sets

Check out the <Expand> button in the middle of the Clause Selection Screen. In addition to allowing you to seeing the clauses that the set comprises, you can use the feature as a set editor.

1. Select set in the Clause Selection Screen;
2. Expand the set;

3. Add or remove terms;
4. Click the More>> button and check the Create Clause-set button;
5. Press **Next>>**.

19.7 Notes, Hints and Tricks

In the examples provided, the clauses were built primarily from one book (observe in the immediately preceding example the 'cntr' prefix, suggesting the same book). A 'fully qualified' terms was thrown in for illustration purposes to show that you can easily mix and match terms from different books (glossaries and/or folders) within a clause-set.

You may call in as 'shorthand' any term contained in any of your books or any of your [SuperBooks](#)^[90]. See [Order of Search](#)^[95] rules.

Final notes re: the "set_" prefix.

- You can easily display 'clause-sets only' onto the Clause Selection Screen. (That way you are not required to scroll through the individual clauses to get to the collection of sets.) Here are several techniques provided by the program to allow this:
 - When you activate Document Assembly (Libraries & Books) screen, and select a book, one of the options in the large box that will appear is "Show Clause Sets." Select it. This is the fastest way to indicate to Pathagoras that you want to see only clause-sets.
 - Pathagoras can easily be set to automatically filter on clause-sets. In the Document Assembly Settings screen, set the prefix of the book to "set_" (no quotes; the underscore character is mandatory). Books so designated are marked by an asterisk to the left of the book's name in the initial Document Assembly (Libraries & Books) screen.
- The 'set_' prefix is required only if you intend to take advantage of the displays discussed above. If you would prefer not to so name your clause sets, Pathagoras will not mind. However, you should still name your clause sets in such a way that they will appear at the top or bottom of, or otherwise clustered within, the Clause Selection screen display.

➔ Remember, a clause-set (even one that is shown as a table) is a Word document. It can be freely edited and re-edited until it does what you need it to. There are several ways to open the clause-set document for editing, but the quickest way may be this:

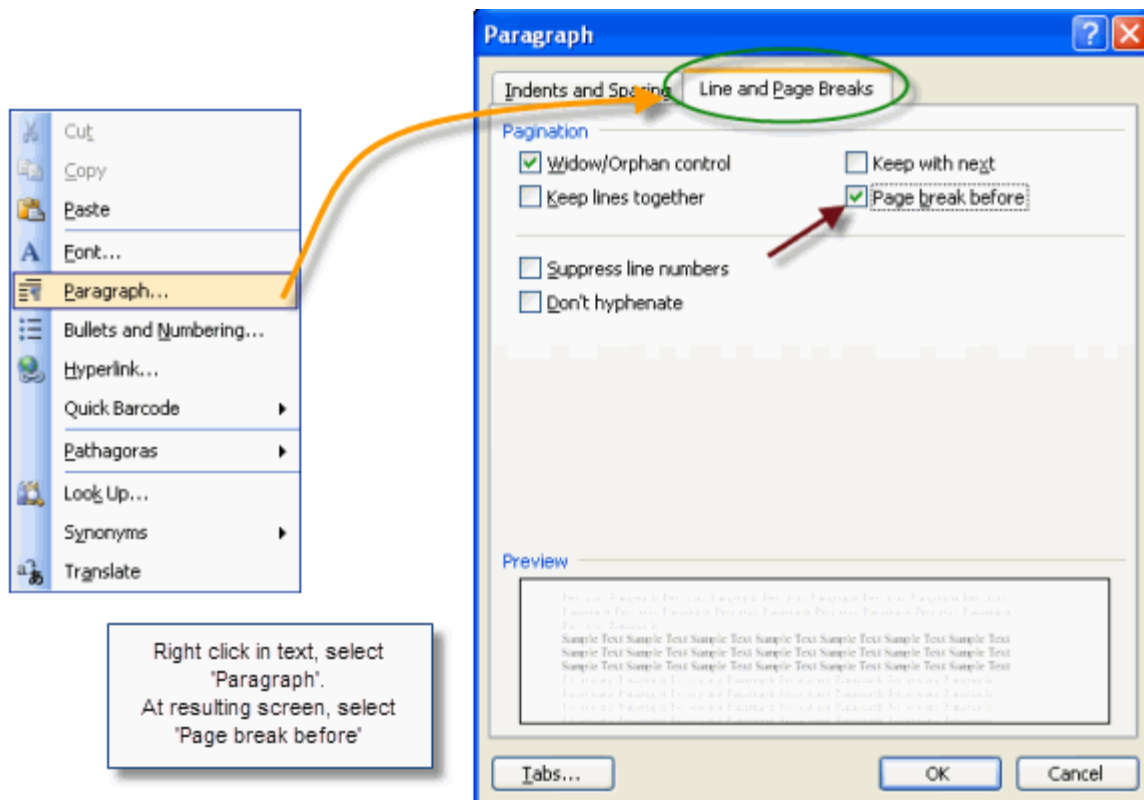
1. Display the Libraries & Books screen.
2. Click on the book from which the set was built and into which the clause-set was saved.
3. Click the 'Open Folder' or 'Open Glossary' choice in the center panel.
4. Locate the set in the folder or glossary. Edit as needed and save your work.

'New Page' Issues

Many users will use Clause Sets (or other multiple document tools) to assemble a 'package' of documents, some or each of which should begin on a new page. The placement of the 'new page' marker is sometimes of critical importance. Most users (understandably) place the new page marker at the end of the earlier clause or document. While there is

nothing necessarily 'wrong' with this approach, in most cases it will insert a carriage return (Enter mark) as the first character of the subsequent document. This may throw off the arrangement of text, tables, etc. of items on the next page.

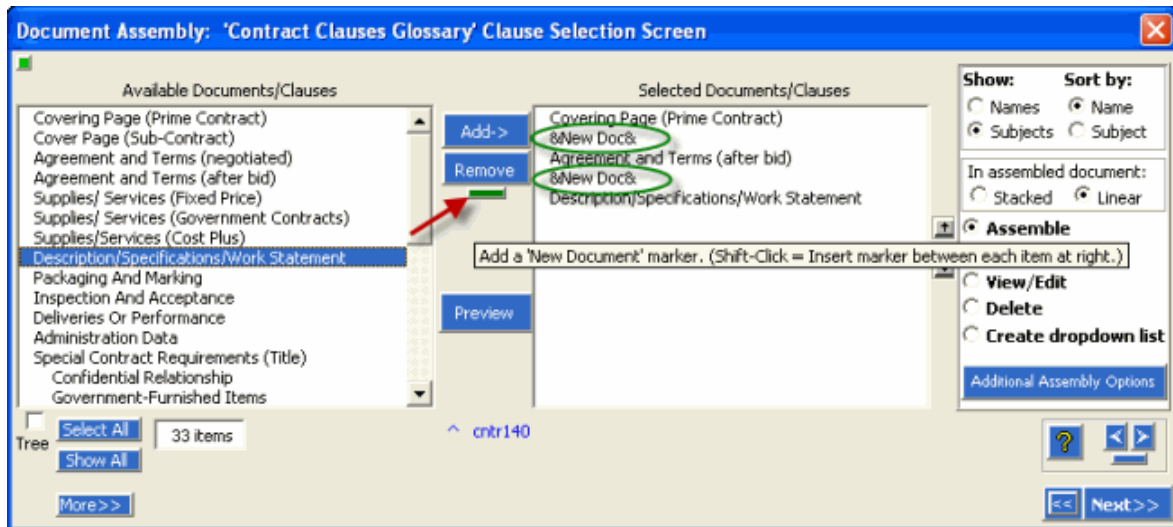
A better approach is to set a 'new page' marker at the top of the 'next' document. Do so via Word's Paragraphs control. Right click in the text that is to be the first line of the 'next' page. Click the 'Paragraph' entry and the 'Line and Page Breaks' tab. Check the box labeled 'Page break before.'



'New Doc'

This is an extension of the above topic, but uses Pathagoras tools to accomplish the goal. Perhaps you want each selection not to appear just on a new page but as a new, separate, document. But initially you want to have all of the text assembled together so that you only have to replace variables one time. This can be easily accomplished.

On the Clause Selection Screen, select each term you want in the ultimate package, but in between each selection, insert a "New Doc" marker. Do so by clicking the green bar that sits in between the two panels, just beneath the Add>> button.



When you click the Next>> button, all selections will initially be one big document, and the text "&New Doc&" will appear at strategic locations (at the point where one selection ends and another begins). You will also see a floating toolbar which reads "Break into Separate Docs." When you have finished replacing variables via the Instant Database, and otherwise finished editing the combined document, click the "Break into Separate Docs" button and Pathagoras will, as expected, break the combined document into its separate sections. The result will be new documents added to your editing screen, each behind a new tab or Word button.

The Pathagoras System

Prefix/Suffix Naming Convention

Part



XX

20 Prefix/Suffix Naming Convention

The way that you choose to name clauses will have a significant bearing on the way you will ultimately use Pathagoras. You should develop a naming scheme that will make your clause names meaningful and easily recall-able.

Pathagoras is flexible enough to work with any of Word's naming styles and rules. However, if you intend to take advantage of Pathagoras' powerful mouseless <Alt-G> method to recall any document from any book, you should use a naming style that allows this to happen.

The 'prefix/suffix naming convention' has been adopted by Pathagoras (and many other programs) as the 'best' way to name clauses in order to implement clause-based document assembly. The following sections explain the convention in detail.

See Also [Prefix Table](#)⁴²⁴

20.1 What Is It?

The Prefix/Suffix Naming Convention is an naming style that allows you to better organize the component clauses of your document assembly systems, and that allows you instant access to clauses named in that style if the prefix has been properly 'registered'.

The Convention uses 'prefixes' to designate a major topic and 'suffixes' to indicate the actual clause.

The prefixes are typically registered in a Prefix Table that Pathagoras maintains. The prefix points to a folder maintained in that table. The result is that you will never have to navigate to the folder containing the clauses again. You would simply type the clauses name. Pathagoras can find the clause by way of the table.

The suffixing numbers allow you complete the naming of each clause (obviously) but more importantly allow you precise control over the ordering of the individual clauses when they display in your [Clause Selection Screen](#)⁵² or [DropDown Lists](#)³⁵⁶.

Structure:

As implemented by Pathagoras, the Prefix/Suffix Naming Convention requires the following elements:

Prefix: two, three or four letters (case does not matter).

Suffix: three or four digits.

Further suffixes: allowed, if the initial 'further suffix' is a letter.

Examples:

These examples meet the prefix/suffix naming convention:

BOC103

pre3433a

dz766first (note in this and the previous examples that the suffix is more that the minimum three or four digits; but remember, the first character after the mandatory suffix must be a letter)

These examples do not meet the convention:

B123 (prefix too short)

clause765 (prefix too long)

100con (prefix must only be letters A to z)

➔ Despite the fact that the three examples immediately above may 'fail' the prefix/suffix convention, they are still perfectly acceptable document names. They simply will not be recognized by Pathagoras for automatic prefix processing. If the document or clause is in a folder or glossary that otherwise will be automatically checked (see e.g., [SuperBooks](#)⁹⁰ and [Search Order](#)⁹⁵), it will be found and inserted.

See also:

[Prefix Registration Table](#)⁴²⁴

[<Alt-G>](#)⁵⁹⁰

[SuperBooks](#)⁹⁰

[Search Order](#)⁹⁵

20.2 Why use it?

There are many reasons why 'prefix/suffix' style names are a preferred naming style for many. Here are a few:

- Prefix/Suffix style names are short.
- A clause or document named following the 'prefix/suffix' style can be instantly recalled from the keyboard. This is so regardless of its actual location. Just type the name of the clause and press Alt-G (for 'g'et).*
- You will have much better control over the order in which clauses will appear in the Clause Selection screen.

Example:

Let's assume doc100 represents the 'Preamble' text to a contract, doc200 the 'Body text,' doc300 the 'Closing Clauses' and doc400 the 'Signature Blocks'. Using the prefix/suffix style, the clauses would appear in alpha/numerical order in the Clause Selection Screen. If the 'conversational' names were used, the alphabetical ordering would be 'Body Text,' 'Closing Clauses,' 'Preamble' and 'Signature Blocks'. In this case, only the last element is in the proper selection order, and that is only by coincidence.

- A more descriptive 'Subject' can be maintained either as a suffix to the actual document name (Pathagoras 2014+) or as a document 'property.' Either way, the user is not dependent upon just the prefix/suffix name in order to know the purpose of the term. Any clause can be instantly displayed on the Clause Selection Screen by either Name or Subject.

- A [checklist](#)⁴²² of names and subjects can be printed for any book. This checklist can be use as a table to match a clause's name with its more descriptive subject.
- You can open the folder associated with the prefix simply by typing the prefix to the screen and pressing <Alt-G>. (Try it: type 'wil' (note-- just 1 'l') to the editing screen and press <Alt-G>. Assuming that you have not repointed 'wil' to another location, you will see the folder containing the Will clauses that shipped as part of the DemoDocs.
- Margin assembly ([described here](#)⁴²⁶) is possible. You have never experienced the flexibility of document assembly via Pathagoras until you have experienced 'margin assembly.'

There really is no 'downside' to adopting the prefix/suffix style. There is no loss of function, only improved document assembly capabilities.

*For the Alt-G recall and 'margin assembly' features to work, you must first associate (simply by pointing) the prefix portion of the term with a folder or glossary. Follow the [steps outlined on this page](#)⁴²⁴ to accomplish this.

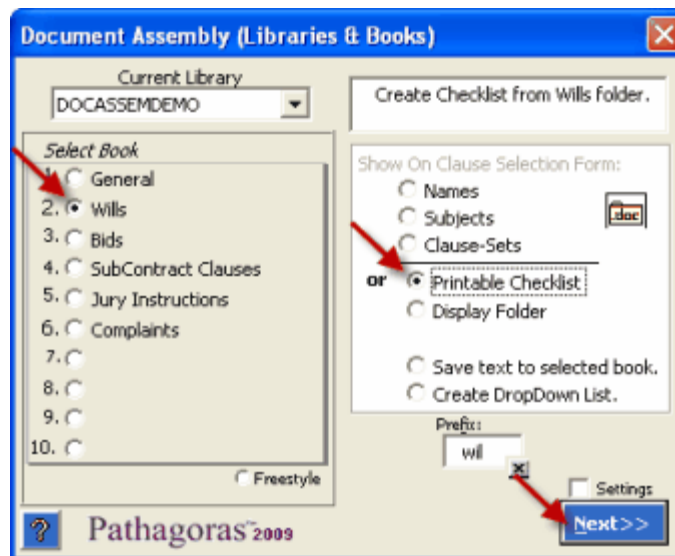
20.3 Printing a checklist of names and subjects.

Pathagoras understands that a name like dec232 means little to the unfamiliar user. However, all term names can be augmented by a subject. The subject can be shown on the Clause Selection Screens.

Further, the user can easily produce a printed checklist showing each term's 'name' along with its descriptive 'subject'. (Even if you have not assigned a subject, you will benefit from a printout of your source clauses.

Creating a checklist: Assuming that you have assigned the folder or glossary to a document assembly book:

- (1) Display the Libraries & Books screen.
- (2) Select the book for which you want to prepare the checklist from the left hand column.



- (3) Click **Next>>**. Answer the prompts which appear asking if you just want 'names only'

or 'names and subjects' to be reflected on the list. Choose names and subjects.

The final product is a checklist showing the term names at the left and the descriptive subjects at the right. Here is what a checklist might look like:

**Checksheet for the 'C:\Program Files\Pathagoras\DEMODATA\WILLS' folder
(C:\Program Files\Pathagoras\DEMODATA\WILLS)**

Name	Subject
<u>wil0120</u>	Administration language
<u>wil0130</u>	Encumbered Property
<u>wil0140</u>	Death Taxes
<u>wil0150</u>	Disp--Jointly Owned Pers. Prop
<u>wil0160</u>	Disp--Tangible Personal Prop.
<u>wil100</u>	Preamble
<u>wil1000s</u>	Complete-Spouse, no children
<u>wil110(m,c0)</u>	Family--Wife, 0 children
<u>wil110(m,c1)</u>	Family--Spouse, one child
<u>wil110(m,c2+)</u>	Family--Spouse, 2 or more children
<u>wil110(s,c0)</u>	Family--Wife, 0 children
<u>wil110(s,c1)</u>	Family--Spouse, one child
<u>wil110(s,c2+)</u>	Family--Spouse, 2 or more children
<u>wil120</u>	Administration language
<u>wil130</u>	Encumbered Property
<u>wil140</u>	Death Taxes
<u>wil150</u>	Disp--Jointly Owned Pers. Prop
<u>wil160</u>	Disp--Tangible Personal Prop.
<u>wil170</u>	Disp--Residue

The resulting checklist is a simple Word document and you can further edit and format it to meet your office needs.

Practical use of a checklist in an office setting:

Original composition: We suggest that you create the checklist following the above instructions, but that you further edit it to provide a more "office forms" feel and consistency. You should add blanks for a client or customer's name to be written at top and other blanks where additional information or notes can be hand written for the operators use.

You, as the author of the ultimate document, should have a supply of these checklists on hand. When you need to create a document for a client or customer, take one of the checklists and check-off a 'first draft' of the document, filling in appropriate information or instructions for the operator.

Pass the completed checklist to the operator who can assemble the document.

Editing: If you later want to add a particular clause into the middle of the assembled and presumably now printed document, and using the checklist as a guide, pencil in just the name of the term at the insertion point. The computer operator would recall the document to the

editing screen and type the clause name at the 'electronic' insertion point, followed by <Alt-G>. The clause will be instantly inserted where typed.

20.4 Prefix Table

Pathagoras maintains a Table which maintains and displays the association of each registered prefix with a specific folder or glossary. By referencing this Table during document assembly sessions, Pathagoras can quickly access all clauses or documents which are named using the [Prefix/Suffix Naming Convention](#)⁴²⁰.

The registered prefixes also allows the user to quickly and effortlessly locate any prefix/suffix named term with a simple press of <Alt-G>. Such 'mouseless assembly' is a hallmark feature of Pathagoras.

The Table is automatically augmented or modified:

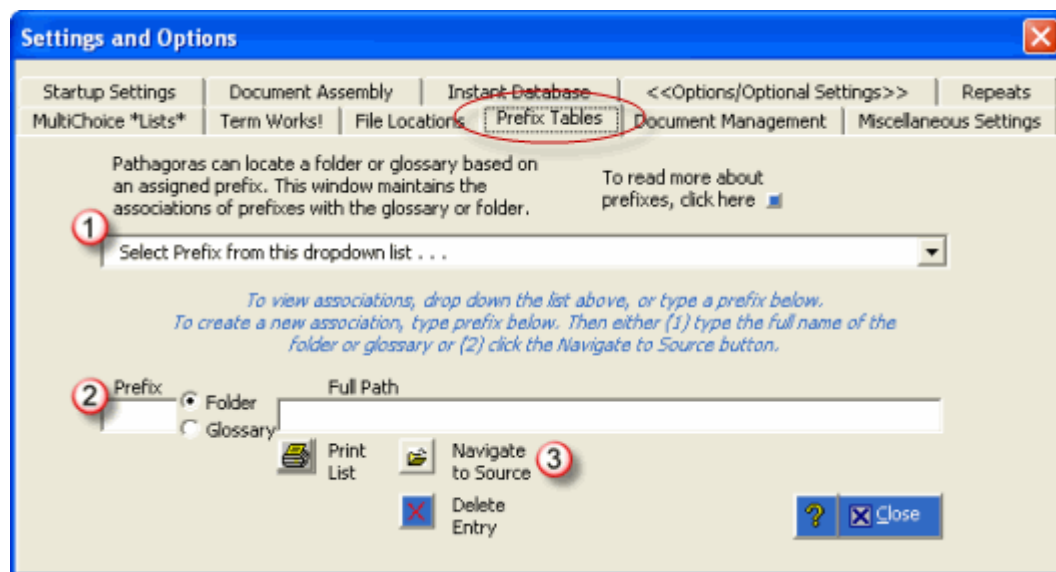
1. when you create a new book and choose to name the terms using the prefix/suffix style; or
2. when you assign a prefix to a book via the Document Assembly Setting screen ('Prefix' tab) and press the Assign Prefix button.
3. when you directly register a prefix using the Prefix Table screen (see below instructions for direct registration).

Direct registration:

To register a prefix and associate it with a specific folder:

1. Display the Utilities/Settings screen and click the **All Settings** button.
2. Click the Prefix Tables tab.

This is the initial view of the Prefix Table screen:



From this screen, you can:

- drop down the list (1, above) to view all currently assigned prefix-to-folder (or

prefix-to-glossary) designations and make changes as appropriate, or

- type into the Prefix box (2, above) the 2 to 4 characters of the prefix. To pair up a folder or glossary to the prefix, click the Navigate to Source button (3, above). Follow the prompts.



If a document bearing the desired prefix happens to be 'on-screen' (and you can easily make that 'happen', an additional button called 'Use folder of underlying document' will appear. Click it, if appropriate, to avoid having to navigate.

20.5 Prefix 'Parsing'

When you type a clause name followed by <Alt-G>, Pathagoras first determines whether the text to the immediate left of the <Alt-G> press is a number or is text.

(1) If it is a number between 1 and 12, it opens the *SmartPath* associated with the number. Pathagoras displays a standard 'File Open' screen and you can open a document shown there, or navigate to another folder.

(2) If the first character of the text is a number between 1 and 12 followed by a colon (e.g., "2:Johnson Will"), Pathagoras assumes that you are calling for a document in a *SmartPath*. Pathagoras converts the number to the associated *SmartPath*. It then locates and opens the document named to the right of the colon. (Please note: this technique should only be used to recall original documents for editing, not for document assembly purposes.)

(3) Otherwise, Pathagoras checks to see if the adjacent text follows the prefix/suffix naming convention:

- First, it checks to see if the name begins with 2 or 3 or 4 letters.
- If it does, it checks to see if the initial letters are followed by 3 or 4 numbers.
- If the prefix/suffix pattern is detected, Pathagoras parses out the prefix and, using the [prefix table](#)⁴²⁴, determines the location of the book associated with the prefix.
- Looking inside the book, Pathagoras locates the specific term and inserts a copy of it onto the editing screen.

All of the above steps are done 'blink of an eye' quickly. (Any time a search fails, Pathagoras implements other logic, discussed in other sections, to locate the requested item.)

(4) If the prefix/suffix pattern is not discerned, or if discerned, but no table entry for the prefix is found, Pathagoras looks for the term following the Search Order rules.

See [SmartPaths](#)⁵³⁶

[Search Order Rules](#)⁹⁵

[<Alt-G> calls](#)⁵⁹⁰


20.6 What if I change my mind?

Let's say you start with the prefix/suffix naming style, but decide that it is not what you really need. Or you decide, after naming an entire book with 'conventional names,' you wish to rename

them following the prefix/suffix style. Not a problem. You can go back and forth almost at will. Two tools will help. See the following:

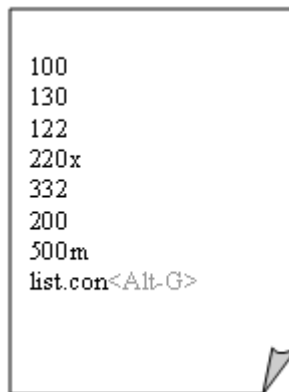
[The Names & Subjects Editor](#) 

20.7 Margin List Assembly

Pathagoras offers an additional document assembly tool designed especially for systems using the [Prefix/Suffix Naming Style](#) . All you need to do is to type a list of the suffixes of the desired clauses down the left margin of the page. Complete the list with the word "list" followed by a "." (dot) and the prefix for the system.

In the example below, the prefix is 'con' for 'contracts'.


Example:



```
100  
130  
122  
220x  
332  
200  
500m  
list.con<Alt-G>
```

Press <Alt-G> immediately after the closing line to assemble the designated clauses. Pathagoras will read up the list until it either finds a blank line or the top of the page. It will assign (in the background) the prefix (in this case, 'con') to each suffix. It will find each clause in the glossary or folder associated with the prefix.

NOTE: It actually doesn't matter whether 'list' precedes or follows the prefix. So long as 'list' is in the closing line, Pathagoras will properly process it. So, in the above example, either 'list.com' or 'con.list' would have been appropriate.

Of course, some will say "How on earth will I ever know the suffixes to type into the list?" The answer is found at under [Printing a Checklist](#) .

The Pathagoras System

Names and Subjects Editor

Part



XXI

21 Names and Subjects Editor

If you want to change the names of all documents in a particular folder, how would you do it? Microsoft allows you essentially one method: display the folder, right click on the file and select "Rename". Repeat for each file you wished to rename. Slow, tedious, error prone. Pathagoras has a tool that highly automates the file renaming process which we call the "Names & Subjects Editor." It is fully described on the following pages.

The Names & Subjects Editor is a very powerful tool that allows you quickly change the names or subjects of all (or several) documents in a selected folder.

The Editor is particularly helpful when you want to:

- edit names to correct misspellings, or globally change a spelling.
- edit names and/or subjects to conform them to a standard naming convention or to add more consistency to the document naming style.
- modify document names and/or subjects to make them more meaningful to you and others.
- adopt the [prefix/suffix naming convention](#)⁴²⁰ for a folder of existing documents.
(Specific steps to help you implement this are found [below](#)⁴³².)
- add other 'prefixes' to documents so that they will appear in DropDown Lists or in the Clause Selection Screen in a desired sort order or to conform them to a alpha-numeric naming scheme adopted by your office.
- modify document names in preparation for a 'folder-to-glossary' conversion.
(Especially important when the current document names do not comply with bookmark naming rules.)

When you activate the Name & Subjects Editor, Pathagoras displays a table containing the current names and assigned subjects of each document in the selected folder. Make changes to one, some or all files listed. Press the 'accept changes' button and Pathagoras makes the name and subject reassignments *en masse* and in a matter of seconds. Need to edit more? Not a problem. Just do it.

Because you are essentially looking at, and can make decisions regarding, the *entirety* of a folder, the power of this tool cannot be overstated.

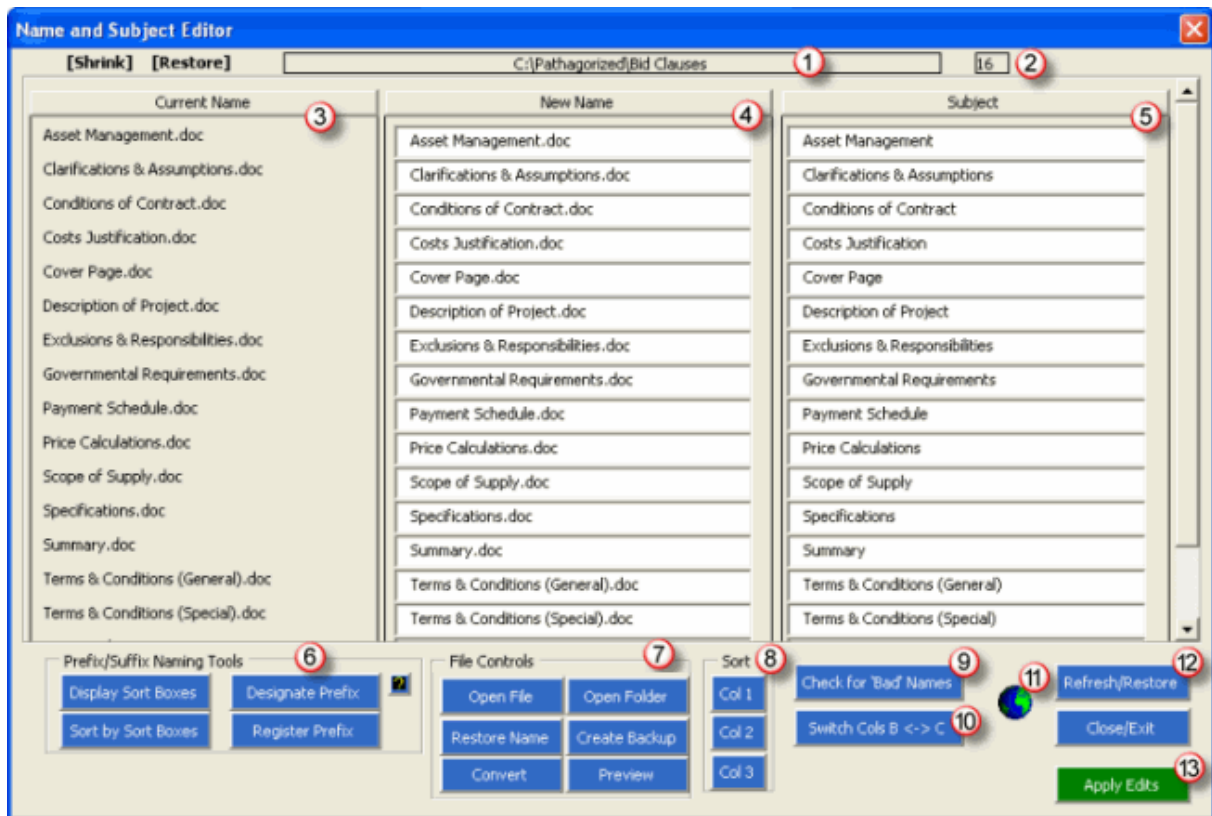
See Also:

[Prefix/Suffix Naming Style](#)⁴²⁰

[Add Subject to Document or Glossary Term](#)³²⁰

[Folder to Glossary Conversions](#)³¹⁸

21.1 Screen Tour



Here are the important elements of the Names & Subjects Editor screen:

1. The selected path.
2. The number of documents in the selected path.
3. An alphabetical list of the names of the documents in the selected path. (This section is non-editable as it is the 'permanent' reference to the old name of the file or glossary term.)
4. An *editable* list of the names. Initially identical to column 1. Make any desired changes to the names in this column.
5. An editable list of the 'subjects' assigned to the document. (The subject of a particular document is found in the document's 'Properties' listing. Pathagoras uses the subject as an alternative to the document's name in its DropDown List and in the Clause Selection Screen.) If no subject was previously assigned to the document, the document's name is repeated, minus the '.doc' or '.docx' suffix.
6. This section provides the tools to help you convert to using [prefix/suffix style](#)⁴²⁰ name. Totally optional. More instructions of converting your document names to this style are provided at [this link](#)⁴³².
7. File controls. A few handy tools that should make navigating your way through the process a bit easier. Among others, you can open the file reflected on the line in which your cursor currently sits, you can open the containing folder, and you can preview the contents of the selected file. (With 'Preview', the screen expands and the first 1,000

characters of the selected clause are shown.)

8. Sort the table by any column you wish. Helpful if you use the files name or subject for sorting its display order.
9. If you have renamed the document, click 'Check for Bad Names' before processing the changes to make sure that the name you selected is 'legal.' A document name may not contain the characters * ? | / \ ; [] < > + = or :.
10. As you are editing, you may find it helpful to switch the Name and the Subject columns. This is the tool by which to do so.
11. Get detailed help on the Names and Subjects Editor (including this page) from the Internet.
12. Restore the screen to its original state. (Pathagoras re-reads the contents of the selected path and redraws the screen with the current state of the folder.
13. Apply Edits: You will be offered the opportunity to make the changes that have been made to just the selected line or the entire chart.

Note: no changes are ever made to your documents until you click "Apply Edits". Therefore you can manipulate the contents of the form until you have things 'just right' before committing to any permanent changes.

21.2 Simple Application

In its simplest application, use the Name and Subjects Editor this tool just to peruse a folder's contents so that you can make minor changes to names or subjects.

Keep the following in mind as you are making your edits --

Naming documents and clauses:

Normal [document naming rules](#)^[60] always apply. If you are working on a glossary (as opposed to a folder of clauses), [bookmark naming rules](#)^[60] will apply. (In a nutshell, a glossary term name must begin with a letter, and contain only letters and numbers (and no spaces). The name may contain an underline ('_') character.)

Subjects:

Unlike a document name (which restricts certain characters from use), you can use any characters (including spaces) for a subject. You should, however, keep the length of a subject to no more than 40 characters. To provide emphasis for a subject, consider adding a special character (e.g., an '*' or (). You could also add spaces to group certain subjects. (Bolding, italics or underlines will not display in the Clause Selection Screen. You cannot use those emphasis tools.)

21.3 Activating the Editor

There are three distinct ways to activate the Editor:

1. from the PathSmart Settings menu,
2. from the Document Assembly Settings menu and
3. from the Editing Tools Screen.

To activate using (1) or (2) above, activate the PathSmart or Document Assembly routines and then click the <Settings> button you see on the main screen.

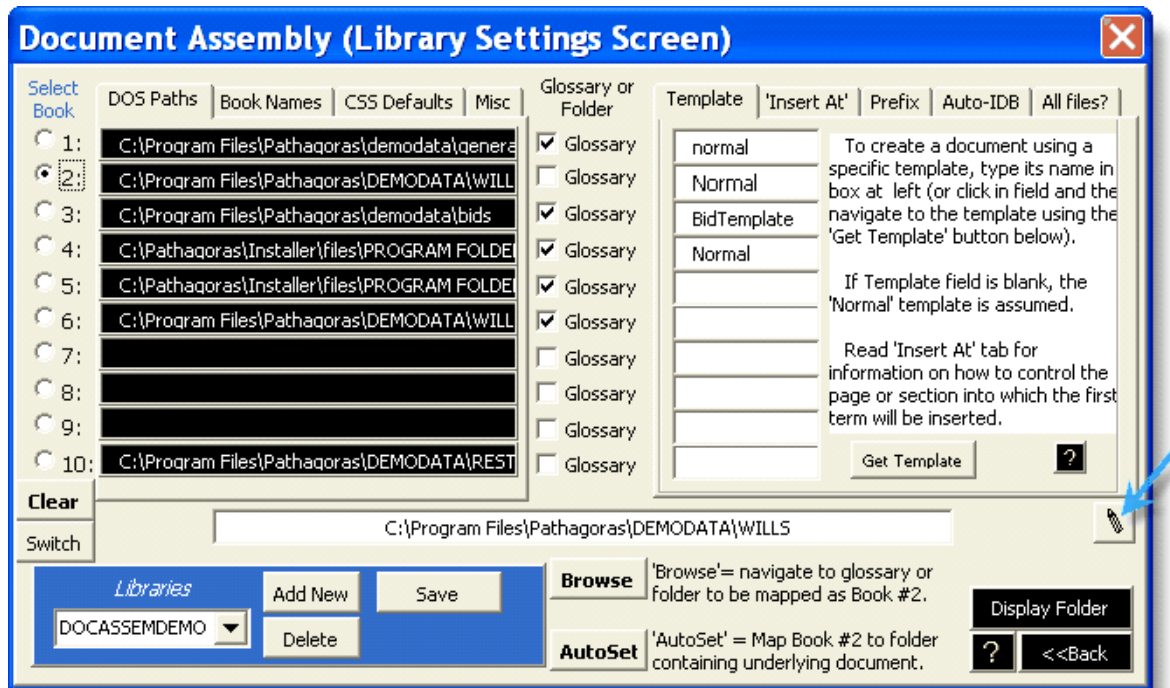

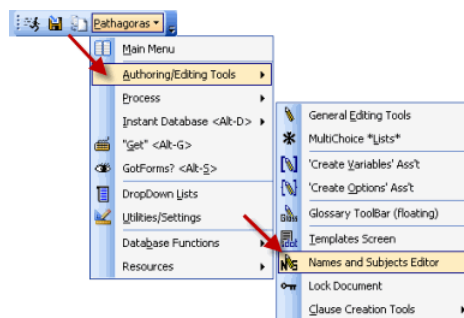


Figure 1 The Documents Assembly Setting Screen

Select a SmartPath or Book you wish to work on. Then, look for the  in the lower right quadrant of the Settings screen. (Figure 1. Note: the pencil button will not appear until you select a SmartPath or Book.) Click it.

To activate the Editor from the Pathagoras dropdown features menu (choice (3) above), click on menu and then click the Authoring/Editing Tools entry and then the Names & Subjects Editor.



You will then be prompted to select navigate to the folder or glossary on which you wish to work.


Pathagoras will display the names and subjects of each term onto a three column form. (See [Screen Tour](#) ⁴²⁹ on previous page.)

- The first column contains a list of the original clause/document names. It maintains the

links to the current document/term names.

- The second columns are for noting the changes you wish to make.
 - Rename the document or term in the second column.
 - Give or rename the document or term's Subject in the third column. (By default, the third column displays the document's current subject. If no subject was previously assigned, Pathagoras will duplicate the document's current name.)

Figure 2 Initial Display

 Note that if you have not previously edited the documents, all three columns will contain substantially the same content.

'Sort' buttons at the bottom of the Editor allow you to sort by any of the three columns to give you a different perspective on how the names might display in other Pathagoras screens.

When you are happy with your new name and/or subject assignments, click <Apply Edits>. Pathagoras will quickly make all of the desired name and subject changes. It will 'repaint' the screen to reflect the new state of the folder or glossary.

You can make even more changes, if desired. Perhaps clauses aren't yet quite in the best order, or a name or subject was incorrectly spelled.


When totally done, close the screen. There is no need to save the table. Pathagoras can quickly regenerate it. (But see "Notes" below for a situation where you may wish to save it.)

21.4 Advanced Application: Prefixes

You can use the Names & Subjects editor to help you to implement the [Prefix/Suffix Naming Convention](#)⁴²⁰ for your documents.

By adopting the prefix/suffix naming style for your documents, you will be able to type the document's name onto your editing screen and then recall the contents of the document into the document simply by pressing <Alt-G>. Don't worry that documents bearing cryptic names such as wil100 or cont345 won't be usable by those who don't know what they contain. That's what the subject field is all about.

Even if you decide to use prefix/suffix naming, before you assign the prefix/suffix names, you may first want to make the list appear in a more logical order. That order should follow the sequence in which the clauses will actually be called into a document.

 Note that when the table first appears, it will display the names of the clauses in alphabetical order according to document name. That may not be the ultimate order in which you want the clauses to appear in the Clause Selection Screen or DropDown List displays. For example, in the below display which showing the contents of a folder containing contract clauses), 'Closing Terms' appears before 'Cover Page.' (Other clauses are also out of 'logical,' as opposed to 'alphabetical,' order.)

To make this happen, you most likely will have to artificially sort the files. Do so as follows:

- (1) Click the <Display Sort Boxes> button in the lower left section of the screen.
- (2) In the new column, type a number reflecting the order in which you want each term to appear.

(3) When you have finished, click the <Sort by Sort Boxes> button. This will cause the entire table will be resorted based on the numbering you provided in the Sort column. (Don't worry for now if you have two 2s or three 5s. You can easily refine the order. Just repeat the steps. When you click "Display Sort Boxes" again, the column will renumber.

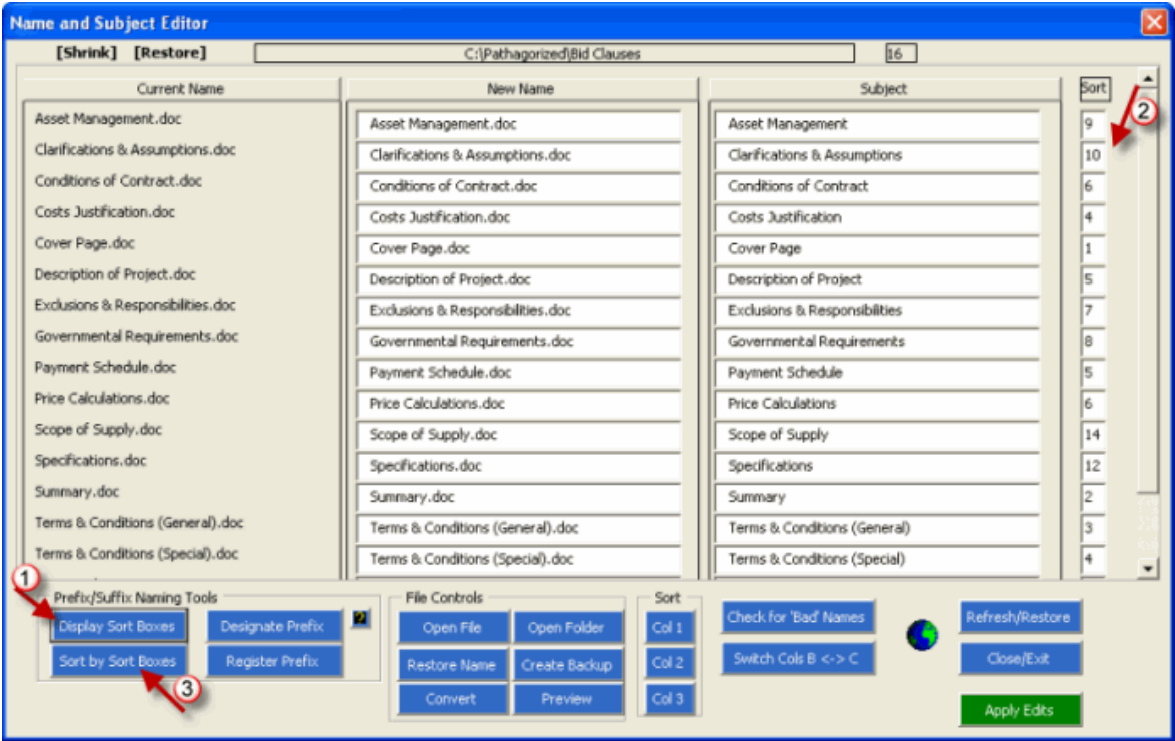
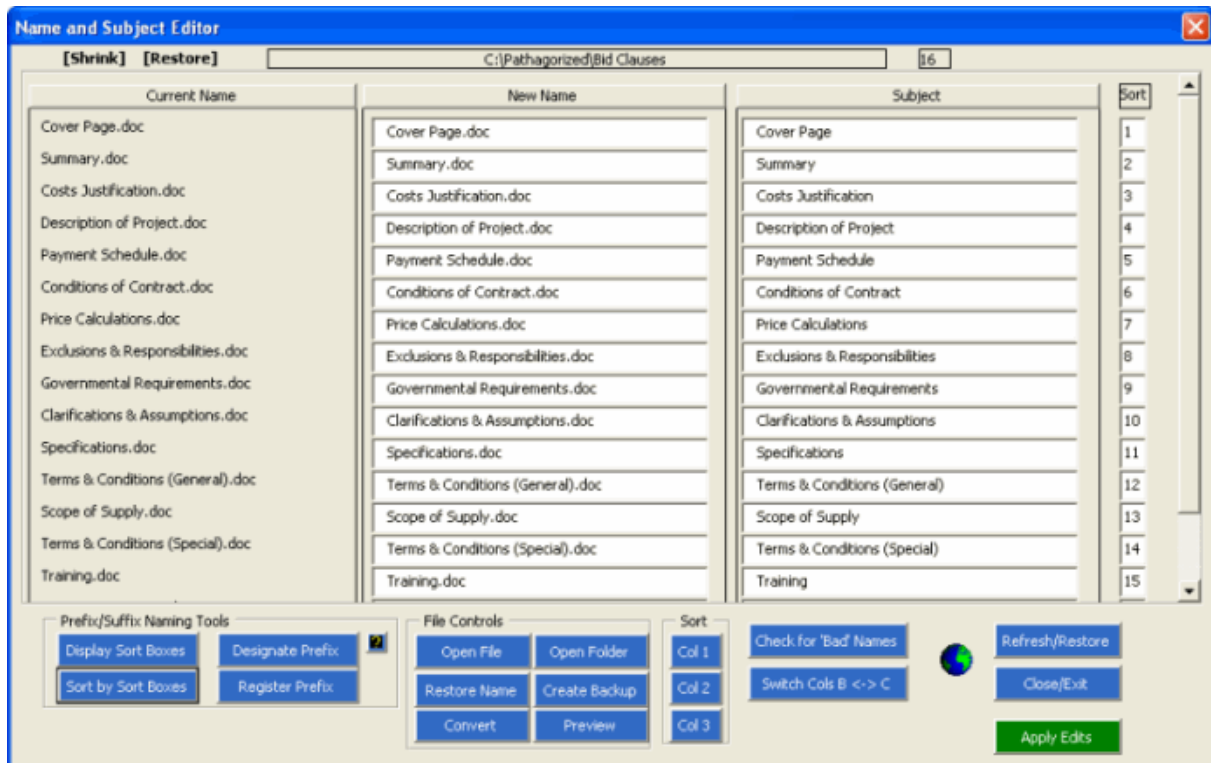


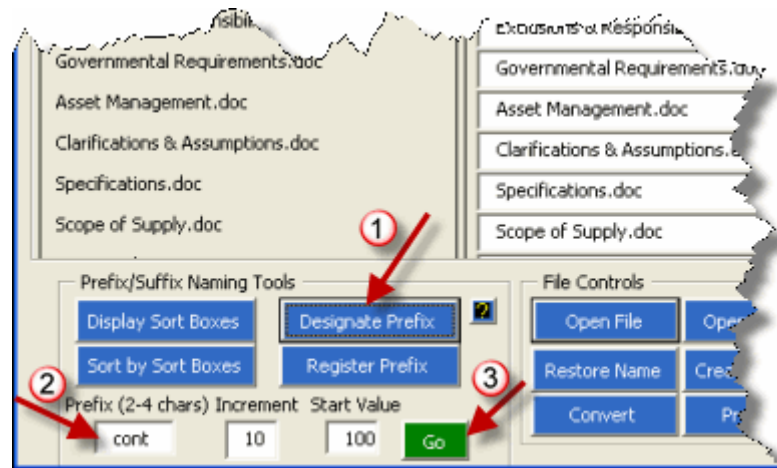
Figure 3. Sorting Numbers Added to right of screen



After the initial sort, if everything is not 'quite right,' you can change the ordering of any element by again editing the Sort boxes and then resort using <Sort by Sort Boxes>. There is no limit to the number of 'sorts' you can perform.

Once the Editor contains the clauses in the proper order, you can rename all of the documents to the prefix/suffix naming style:

1. Click the <Designate Prefix> button. A new set of text boxes will display at the bottom left of the screen.
2. a. Provide the prefix you want the clauses in this folder to be known by. Here we have selected 'cont' for 'contracts'. The prefix can be any combinations of letter, but must be 2, 3 or 4 characters long.



- b. Confirm or modify the Increment (numerical 'space' between document names--you should have at least 5 so that there is room to add more clauses) and the Start Value.

The start value must be three (preferred) or four digits long.

- c. Continue to modify the subjects (third column) as appropriate. No naming rules apply to the third column. Just remember that the subject is what will appear when the user chooses 'Display by Subject' for the Clause Selection Screen or DropDown Lists displays. As a rule (of thumb), however, a subject should not exceed 40 characters.
3. Click the <Go> button (still in the Prefix/Suffix section of the screen) and Pathagoras will convert the names in the "New Name" (second) column to those following the prefix/suffix naming style.



NOTE: CHANGES TO THE ACTUAL DOCUMENT NAMES HAVE NOT YET OCCURRED. You can continue to experiment with document names, prefixes, sort order, etc. until you are satisfied with what you have.

(And even after you commit, you can still make changes. Do not 'fear' this process.)

Once you are satisfied with the names and subject you have selected, click the green <Apply Edits> button.

- If you have not yet registered the prefix to Pathagoras' Prefix Table, you will be asked if you wish to do so now. Say Yes.
- You will also be asked if you wish to create a backup of the original files, using their original names, in a 'backup' folder. You probably should say yes in case you decide to reverse the entire process.
- Then, Pathagoras will begin the process of renaming the original documents (shown in column 1) to those names provided in column 2 and with the subjects provided in column 3. There will be a bit of screen flashing as each document is opened, renamed and saved.
- You are done. Assign this folder to a book or a DropDown List. And remember that you

can recall any document from this folder from now on by typing its name to the editing screen and pressing <Alt-G>.

21.5 Advanced Application: Renumbering

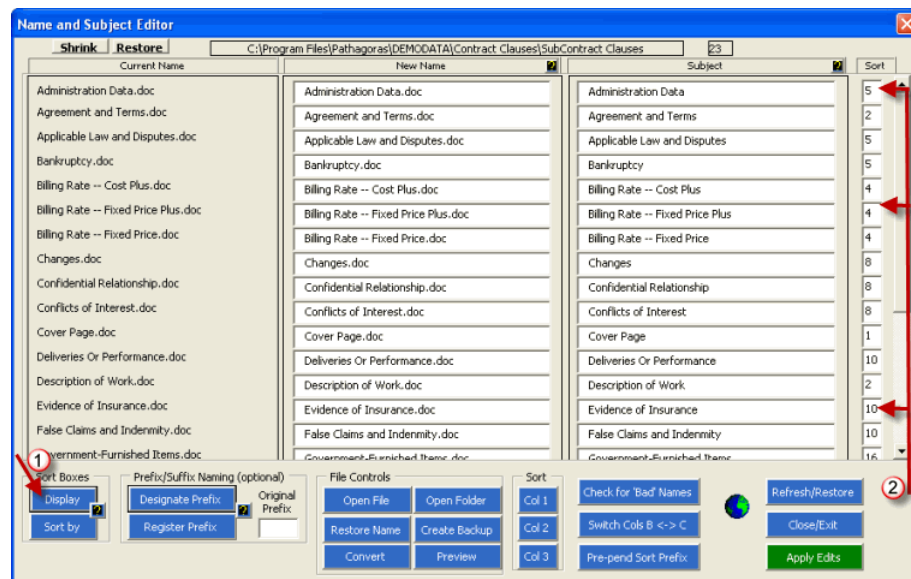
The previous examples used 'Prefix/Suffix Naming' to order your documents. That method requires you to completely rename your documents to the prefix/suffix style. This is not a 'bad' thing (indeed, we strongly recommend it).

But if you are not quite ready to make that jump, there exists another quite satisfactory approach for causing your clauses to appear in a proper 'selection order.' You need only to add 'sorting numbers' to the 'front' of the document names in column 2. With nothing further, you would have a better system than you had before.

Caveat, if you are changing the names in a glossary: Before getting too far with this, if you are renaming terms in a glossary, be aware that glossary terms must be named using the conventions for naming bookmarks (since the individual terms in a glossary are identified by bookmarks). Therefore, the first character of a glossary term name must be a letter A thru z, and cannot contain any not alpha-numeric characters other the underscore character. (The bookmark cannot start with a number or non-alphabetic character.) You should use the prefix/suffix naming style if you want to rename your glossary terms in an ordered fashion.

There are two distinct ways that you can add these sorting prefixes. The first is manually. Simply type "01", "02", etc., to the front of each document's 'New Name' shown in column 2, making sure that the prefixes match the display order you desire. (Note the '0' in front of single digit numbers. This assures proper 'alphanumeric' sorting.)

A more automatic method is to pre-sort the files in the Editor display. You would first use Sorting boxes (shown below) to get the order of your files in the proper top-to-bottom order. Then you would tell Pathagoras to add the proper prefix. The following images illustrate this technique.

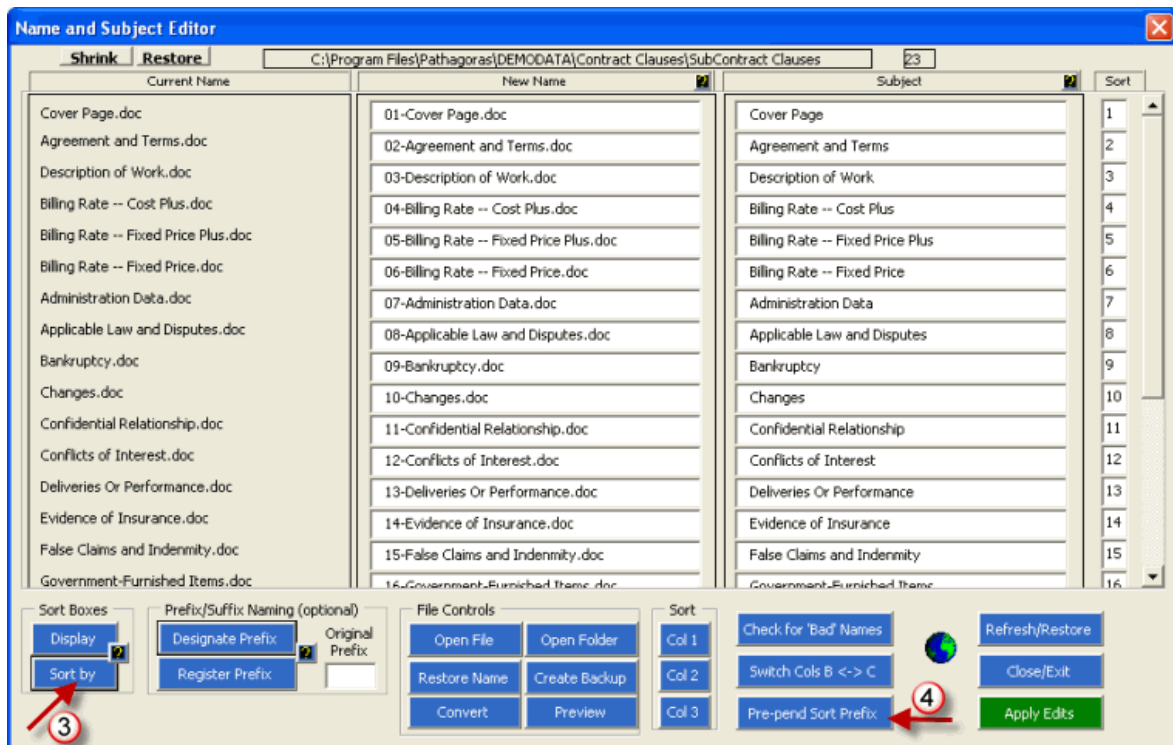


To assign sorting numbers to the front of each document in the 'New Name' column:

(1) Display the Sort Boxes.

(2) In the boxes that appear at the far right, assign relative sorting values to the clauses, with the ones you want to appear at the top with the smallest numbers and those near the bottom with the largest numbers.

The actual sort order does not matter on the first pass. You can sort and refine as many times as needed.



(3) Click the 'Sort by' button (sort by Col 4 in Pathagoras 2013) to see the first round of results.

Repeat (1), (2) and (3) to to refine the ordering.

(4) After you have the desired sort order, click the Prepend Sort Prefix. Answer the formatting questions (space or no space, hyphen or no hyphen). The results will be shown under the New Name column.

Check your proposed results. Edit as needed. When satisfied, click <Apply Edits> (the green button) to rename the files to the 'New Names.'

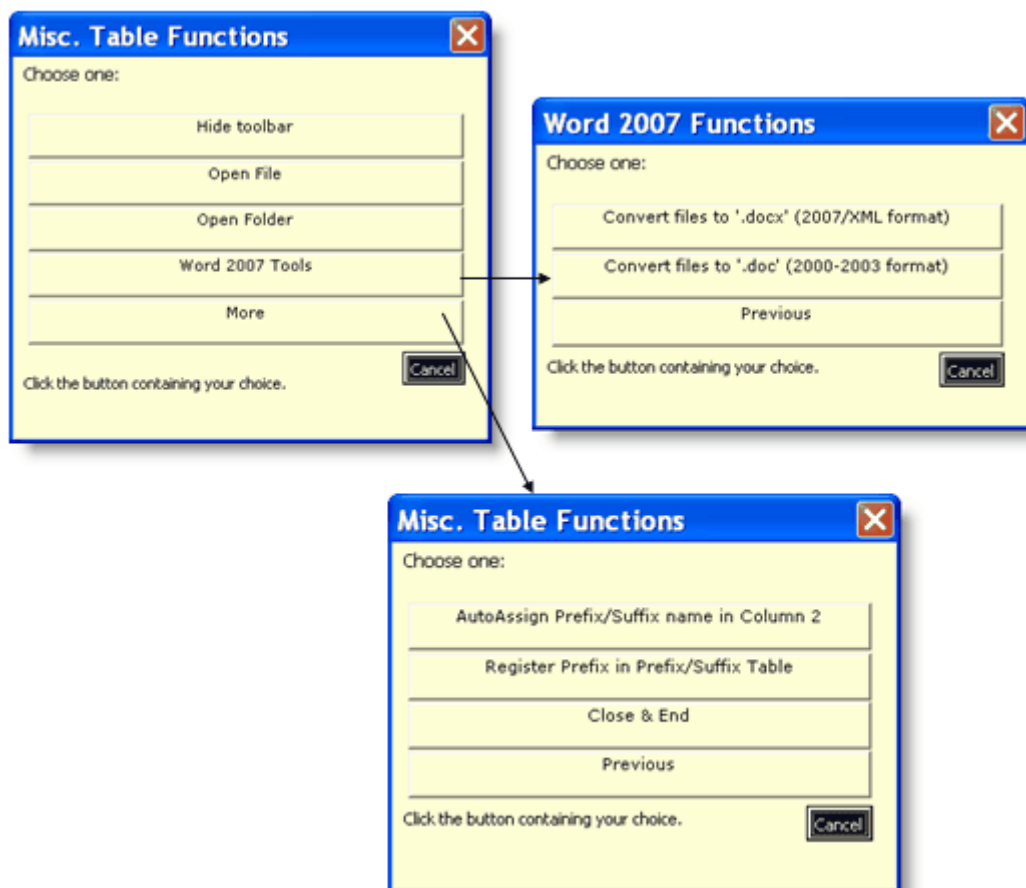


Keep in mind that Word (and Pathagoras) will sort document names in the various available displays alphabetically (not numerically). '1', '100' and '1000' all are 'alphabetically' ahead of '2.' So, to make sure that '2' appears ahead of '10', add a '0' ahead of the single digit, '00' before single digits and a single '0' before double digits if you think your numbering will exceed 100.. That will keep the intended order.

21.6 Notes and Miscellaneous Items

The Miscellaneous button offers quite a few additional features, including conversions to and from the 'legacy' (Word 2003 and prior) and the new Word2007/2010 formats. On the 'front' page you are provided options to

1. Hide the toolbar. (Restore it by right clicking in the 'grey area' in the toolbar area and click the Names & Subjects entry.)
2. Open the file shown in the table cell in which the cursor is resting.
3. Open the folder or glossary that populates the table.
4. Display the Word 2007/2010 options menu.
5. Display addition options.



NOTES:

Do you have too many terms to prepare at one sitting? Not a problem! You can save the Name & Subject Table in the same way that you save any other document. Take it home with you. Edit it there. Bring it back to the office and run the changes a day, a week or a month later. Take your time.

The compressed "all-on-an-editable, printable page" view of names and subjects provides unmatched ease for both renaming clauses and for assigning/reassigning meaningful subjects to your terms. Before, you would have to change a document name

or subject one at time. Now you can do it multiple files at a time.

You can use this feature for any folder that contains Word documents, not just 'Pathagoras' folders.

You do not have to actually rename any documents in order to use the Word2007/2010 tools. Display any folder, jump directly to the 2007/10 screen and you can quickly convert an entire folder of documents to the new XML format. Pathagoras will even offer to backup the original files into a new sub-folder.

Warning:

While you can use this is feature for changing any document names, remember that if you reference the names within a previously created clause-sets, you must change the 'pointers' contained in the clause-sets to reflect the new names. Otherwise, you will get an error message during assembly. It is easy to change the pointers. Just open up the clause-set document (or the glossary containing the sets). Change the pointers using any Word tool.

See [Clause Sets](#) 

22 Right Click Menu

Pathagoras has always provided easy access to its functions for the two basic kinds of computer users: those who prefer mice to perform computer tasks, and those who prefer to invoke as many commands as possible from the keyboard. That's why you can call just about any Pathagoras function using either the keyboard or the mouse.

This feature, however, is for 'mouse fans.' Mousers know that a good set of right-click commands will speed up computing faster than just about anything else. Pathagoras will surely please with this new feature.

Pathagoras 2016 introduces robust, personalized, Right-Click functionality. It starts with our new Right-Click Assignments screen (discussed below). Via that screen, you can:

- assign up to 30 right-click functions help you speed through document editing and other automation features.
- assign up to 10 locations into which you can disassemble your documents. (After you have made your right-click assignments, you can simply highlight a snippet text and, with right-click speed and ease, create a new entry in one of your DropDown Lists or into your Super Folder or Super Glossary.)
- easily designate a set of variables for right-click insertion. (When you are not sure of the precise spelling of your various variables, this can be a real time- and click-saver.)
- make the contents of your Super Folder and of your Super Glossary available for instant insertion into any document. (Here we mean *no navigation!*)
- avoid the need to first click the Pathagoras tab to access the functions you have assigned a right-click menu.

When we initially designed right-click functionality, we pre-assigned all choices. But that turned out to be an instant non-started. So we started over. Rather than or restricting the menu to the ones we thought were 'most important', we made it possible for you to design your own menus. A right-click menu that you can personalize is way better than a pre-set menu.

Getting Started. Designing your Right-Click Menu:

Setting the buttons you want to appear in your right-click menu is done via the Right Click Assignments screen. To access it, click Pathagoras Features | Authoring/Editing tools. Look for the Right-click element.

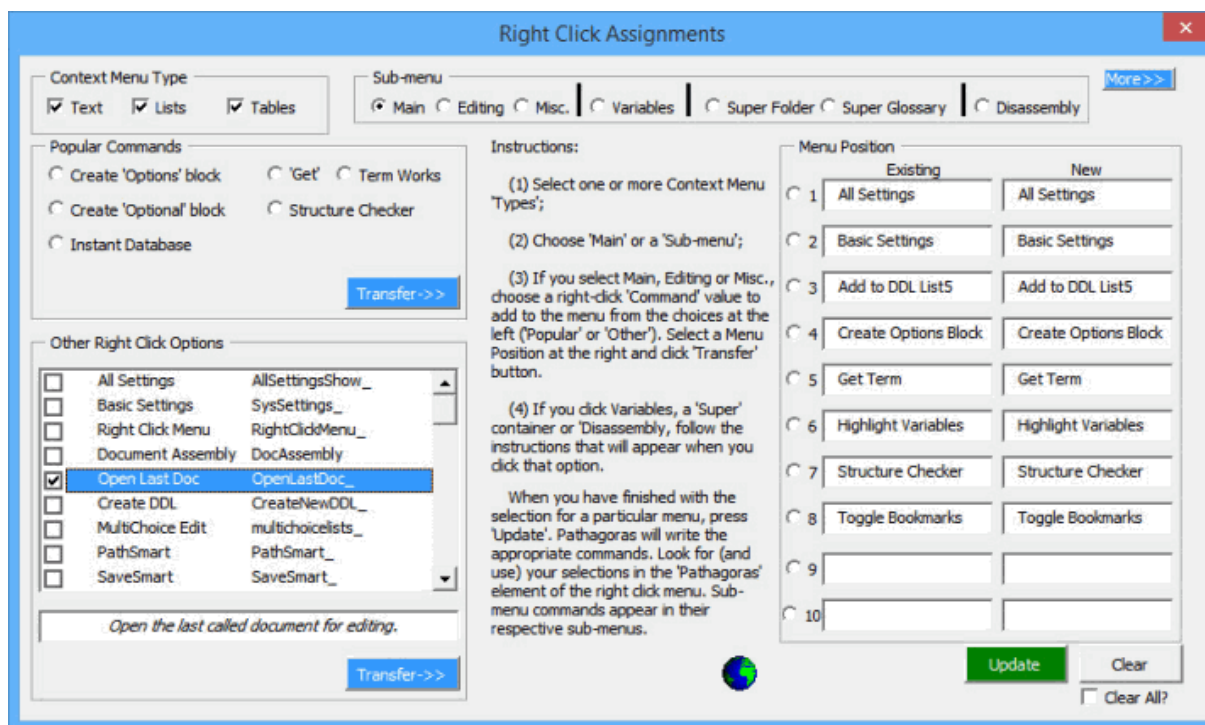
Before you start, it may be a good idea for you to elevate the call for Right Click Assignments screen to the Quick Access Toolbar. You likely will be calling on the Right-Click Assignments screen quite a few times as you build and refine your Right Click menus. The button for the Assignments screen can be found at "Pathagoras Features | Authoring/Editing Tools" (look for the 'Right Click Menu' element. Right click on it and click 'Add to Quick Access Toolbar.')

With that done, you can call the Assignments screen from the QAT quickly and without any navigation.

(You will likely need the Assignments screen frequently while you are assigning elements to the right-click menus, but once they are 'mature', you may never need this screen again. At that time you can remove the Assignments screen element from the QAT.)

Screen Tour:

Main, Editing and Misc.



This is the screen you will see when the 'Main', 'Editing' or 'Misc.' sub-menu choice selected.

Context Menu 'Type':

Microsoft provides dozens of right-click possibilities. (The actual number is 120 separate right-click menus.) Which one appears when you right-click the mouse depends upon the location and context of the cursor at that precise moment you click. (The contexts include clicks on plain text, numbered text, text boxes, images, numbered lists, spelling errors, etc. This is why Microsoft has also accurately dubbed the feature the 'context menu').

It simply is not possible to add a Pathagoras right click tool to every possible right-click context menu. We initially provide connections to the most popular 3 (Text, Lists and Tables). Not to worry: we will be adding more as this feature matures and customers tell us what they need. All three are checked by default, but if you know a particular command is not needed in a particular context, you may uncheck it.

Sub-Menus

You can assign up to 10 right-click commands to the 'Main' level ('Pathagoras') right click menu *and* into to each of the sub-menus titled 'Editing' and 'Misc.' (The 'Variables', 'Super Folder', 'Super Glossary' and 'Disassembly' sub-menu options are discussed in the sections further down this page.)

Menu Positions:

When you click one of the first 3 Sub-Menu type, a series of numbered boxes will appear at the right side of the screen. These boxes contain (or will contain) the Pathagoras command you wish to assign to your right-click menu. Currently you can assign up to 10 commands for each Sub-Menu type. (Note: 'Main' is actually not a 'sub'-menu. It's the top level in the right-click menu labeled 'Pathagoras'. All others, however, will fall under the Pathagoras menu as sub-

menus.)

The order of commands in the Menu Position display will be the order in which the commands will appear in your right-click menu. You can set any order you wish: alphabetical, frequency of anticipated use, etc. But you must set the order manually.

Command List:

The left side of the screen displays the various Pathagoras commands that you can assign to a right-click menu or sub-menu.

- The upper section contains a few of what we have dubbed 'Popular' commands. This listing is entirely arbitrary on our part. Let us know what else you think should be in this list.
- The lower section are the other commands that can be inserted as right-click options.

These are the same ones you can assign to an [Alt-Q 'My Button'](#) ¹⁸².

To assign a command to a button, click on one of the Menu Position buttons (left side) and then select a command you want to be placed in that position. Then click the Transfer button. (No blank lines are allowed. So if you try to save a command to #9 in the above example, you will get a process error.)

Nothing is written out at this point. You can change your mind, and reorder your commands to your heart's content.

Update:

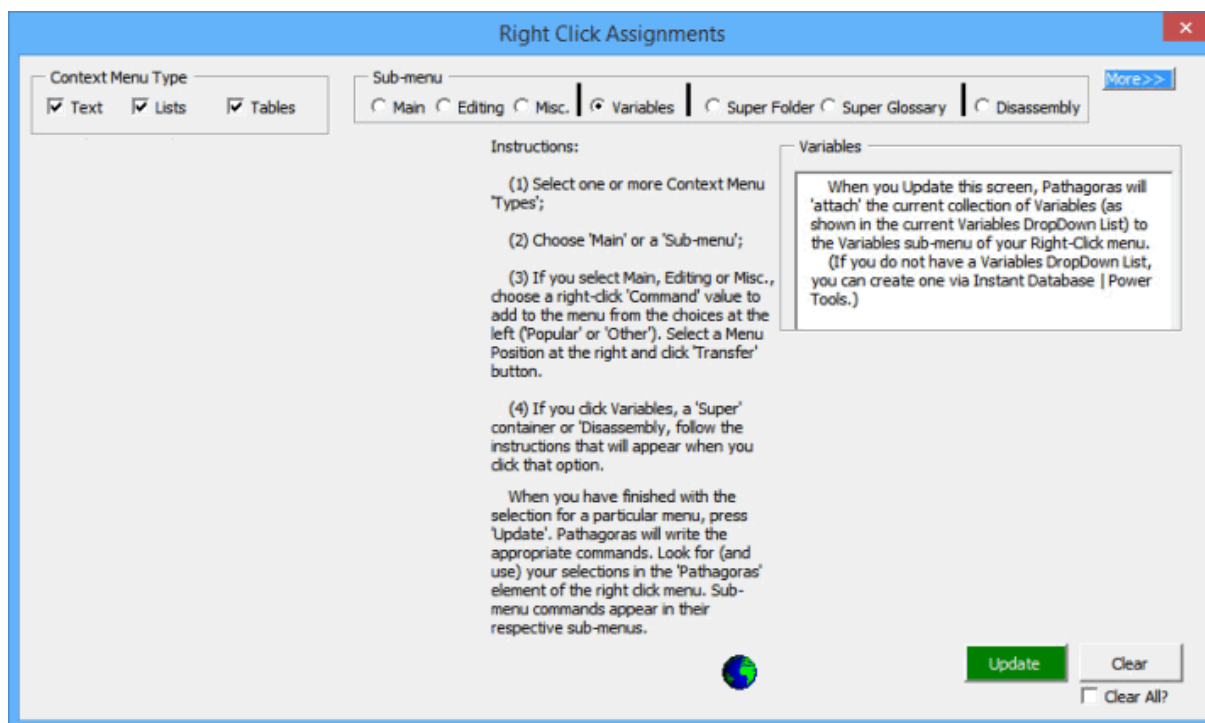
When you have finished with your selections and ordering, click the Update button. (Note: You must 'Update' each sub-menu page one at a time.) Pathagoras will quickly create the right-click menu for the displayed level from the selections made.

Testing: To test things out, close the Right Click Assignments screen. Then right-click on some text. The Pathagoras menu will appear at the top of the context menu. Select an item to experience the right-click action.

Clear & Clear All: If you see your lists duplicating on otherwise not making sense, Clear or Clear All and start over. You do not have to worry about losing any of your previous selections. All the commands you selected in the last session when you pressed Update was preserved. You will just have to re-click the Update button for each screen after you Clear All. Clearing and resetting take mere seconds. Do not hesitate to edit and refine your selections.

Variables:

When you click "**Variables**", the screen will change to show a message that the press of the 'Update' button will insert your collection of variables as shown in the DropDown List bearing that name into the right-click sub-menu titled 'Variables.' From then on, when you click on a variable shown in the 'Variables' right-click menu, that variable will be inserted into your document. This is more than cool.



This is the screen you will see when you select Variables from the Sub-menu listing at the top.

Of course, you must have previously created a Variables Drop Down that contains the variables you want also to be right-click available. The steps on how to create this DropDown can be found [at this link](#)³⁷².

Super Folder and Super Glossary:

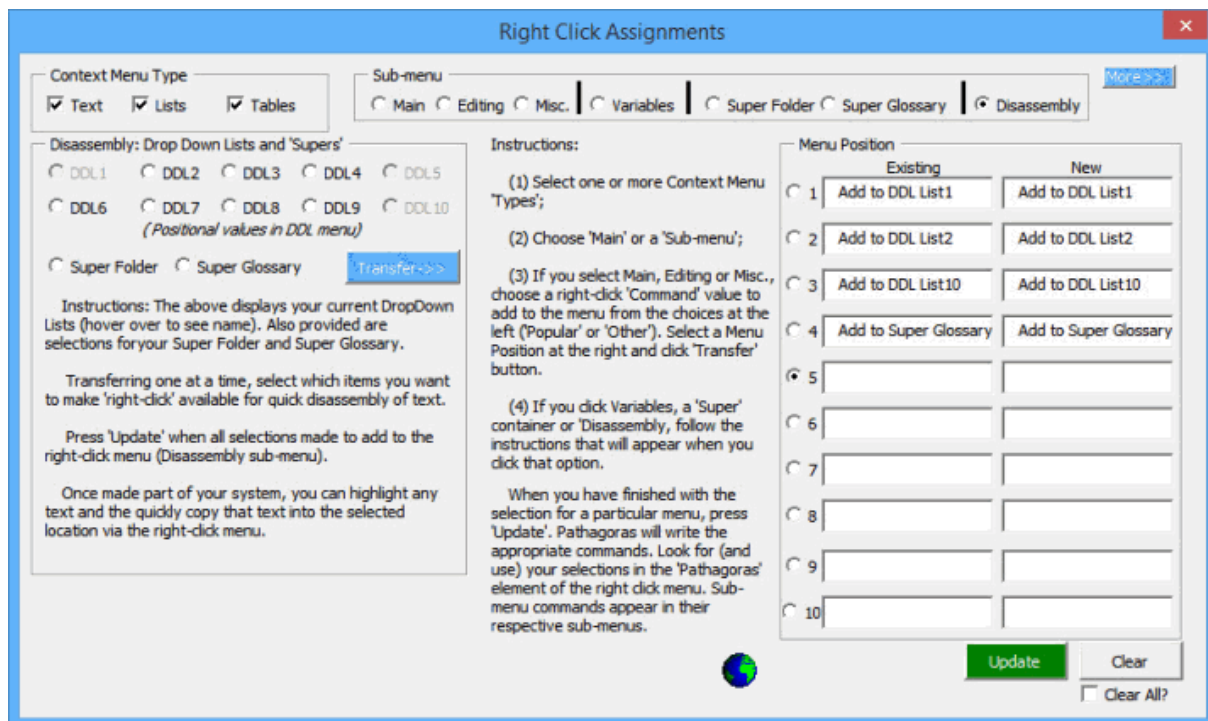
Choose one of these options and the screen will change to show a message that the press of the 'Update' button will insert your collection of documents and terms into your right-click menu. Then, when you click on a button in your right-click menu, the term assigned to that button will be instantly inserted into your document. No more navigation to get to common terms, phrases and even whole documents!

Of course, you must have previously pointed Pathagoras to a Super Folder or Glossary. The steps on how to do so can be found at these links: [Super Folder](#)⁹⁰ [Super Glossary](#)⁹¹

Disassembly:

The right-click menu was originally conceived by us as a document assembly tool. Right click . . . insert text . . . right click . . . insert text. But users who frequently disassembled text using DropDown Lists and the Alt-G function proposed that the right-clicks be used to enhance those document dis-assembly function as well. So we obliged. Now it is even easier to create libraries and folders containing your building blocks of primary documents.

When you click the 'Disassembly' option, the screen will change to display the various locations into which you likely would want to disassemble your documents. (We assume those locations to be your DropDown Lists and your Super Folder and Super Glossary. If we have guessed wrong, let us know.)



This is the screen you will see when you select 'Disassembly' from the Sub-menu listing at the top.

The Disassembly screen lets you assign up to 10 locations that (when you call on the right-click menu) you can quickly select for receiving highlighted text. The left side displays your (up to 10) DropDown List folders as well as the Super Folder and Super Glossary you may have assigned. Just select and 'Transfer'. When done, click 'Update.'

Once done, try it out. Highlight a snippet of text you want to disassemble from the main document. Right click and select the 'Disassembly' sub-menu. Then select the target. Pathagoras will ask for a name for the new document snippet. Provide the name and press OK. The highlighted term will be instantly saved as a new document in the desired location. (If you are saving to a DropDown List and it doesn't immediately appear, click the <Refresh> button in the List.

Your Help Requested: This feature was first released in early July 2015. Being a brand new feature, we are not quite sure the extent to which users will implement this tool, and how. We anticipate it will be used for common tasks, but could just as easily be used for 'remembering' rarely used features. We have included a list of many of Pathagoras features, but we know that not all are included. We are working on that. And we will need your help. Your comments, complaints and suggestions and wishes are most welcomed. Write to us about the design, the included features, the ones we forgot, etc., at info@pathagoras.com with any words you wish to share.

The Pathagoras System

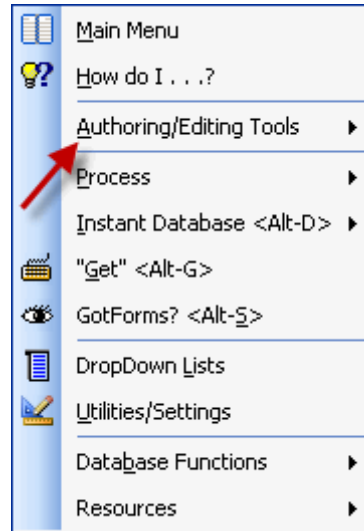
Miscellaneous Editing Tools

Part

XXIII

23 Miscellaneous Editing Tools

Pathagoras provides a variety of tools to make editing your documents easier and more accurate. Access the Editing Tools via the Pathagoras dropdown menu.



23.1 Balancing Markers

Error messages received during document assembly are almost always caused as a result of an imbalance between the opening and closing markers. The more [bracketed variables] and <<*Options/ Optional*>> text blocks you add to your documents, the greater the possibility of an editing error.

Pathagoras can help you find such imbalances and can even direct you to their likely locations.

1. Display the 'Authoring/Editing Tools' menu and then the 'General Editing Tools' sub-menu.
2. Click 'Check Structure.' Pathagoras will automatically count each marker typically used in Pathagoras and report back which ones are out of balance. Upon request, Pathagoras will also check other 'structure' elements in your document and report any problems it discovers. See "[Structure Checker](#)"¹⁷³ for information on this powerful tool.
3. When you 'run' a document, if a document structure error is detected, Pathagoras will typically announce the cause of the error and highlight the text that has created the error.



As you are testing your documents prior to releasing them to the end users, you might want to consider the Editing Tips found at the end of this Chapter. Using those tips, you can more quickly isolate the section of text causing the problems.

23.2 Color Coding (Paint Markers)

Once you work with variables and with <<*Options/ Optional*>> text blocks for a while, you may wish some visual assistance in seeing the various elements of the document.

1. Display the 'General Editing Tools' screen.



2. Select the 'Paint' button. This will 'paint' the various brackets and braces in different colors so that you can more easily identify the various elements in each text block sections.

This same feature is available:

- under the 'Wizards and Assistants' element of the Pathagoras dropdown features listing and
- as one of the default [MyButton](#)¹⁸² (Alt-Q) buttons. (If not currently set as a MyButton, it is easy to do so. Select the 'Paint Markers' element.)

23.3 Accessing Source Clauses

Pathagoras provides quick, direct access to your clauses that are reflected in your books or in DropDown List. You will not have to navigate to the source clauses.


To access clauses in a 'book'--

1. Display the Document Assembly (Libraries & Books) screen.
2. Select the book containing the documents you wish to edit (here, be sure to not double click the selection).
3. From the white box that appears when the screen expands, select "Open Folder" (or "Open Glossary").
4. Press **Next>>**. A standard Word/Windows file open screen appears.

5. Locate the document you wish to edit, and edit away.

To access clauses in a "DropDown List"--

1. Scroll to the bottom of the list containing the document you wish to edit.
2. Look for and click the entry that reads <Open Folder> (or <Open Glossary>).
3. A standard Word/Windows file open screen appears.
4. Locate the document you wish to edit, and edit away.

 Your source clauses are stored as regular Word documents. That being the case, you can also access them using standard Word and Windows navigation techniques. A change to a document using Pathagoras tools or a change to a document using standard Word/Windows tools always yields the same results. Whether using Pathagoras tools or not, you always in Word and all Word editing tools and rules apply.

23.4 Search & Replace

Perhaps you wish to change a variable name from [Name of Client] to [Client Name]. Perhaps you want to change the enclosing brackets from '[' and ']' to something else.

Perhaps you want to change the word "Executor" as it appears in your estate planning documents to "Personal Representative."

Making changes such as these within a single document is pretty easy. Just use Word's Find and Replace function and you can accomplish the task in short order.

But if [Name of Client] appears in dozens of document, and dozens of places within each document, you will want to make those changes using a tool than provides a more global scope. Pathagoras gives you 2 pretty powerful tools. Its built in [Search and Replace](#)⁴⁴⁸ tool and one that may surprise you -- the [Instant Database](#)⁴⁵¹.

Check out the next two sections for more information.

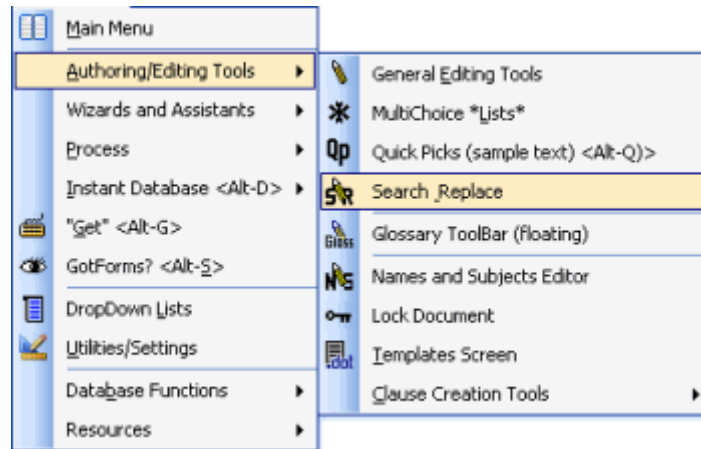
23.4.1 Search and Replace Form

Pathagoras lets you replace selected text in the active document and the same text in all other documents that reside in the same folder. Its operation is similar to that of Word's search and replace function.

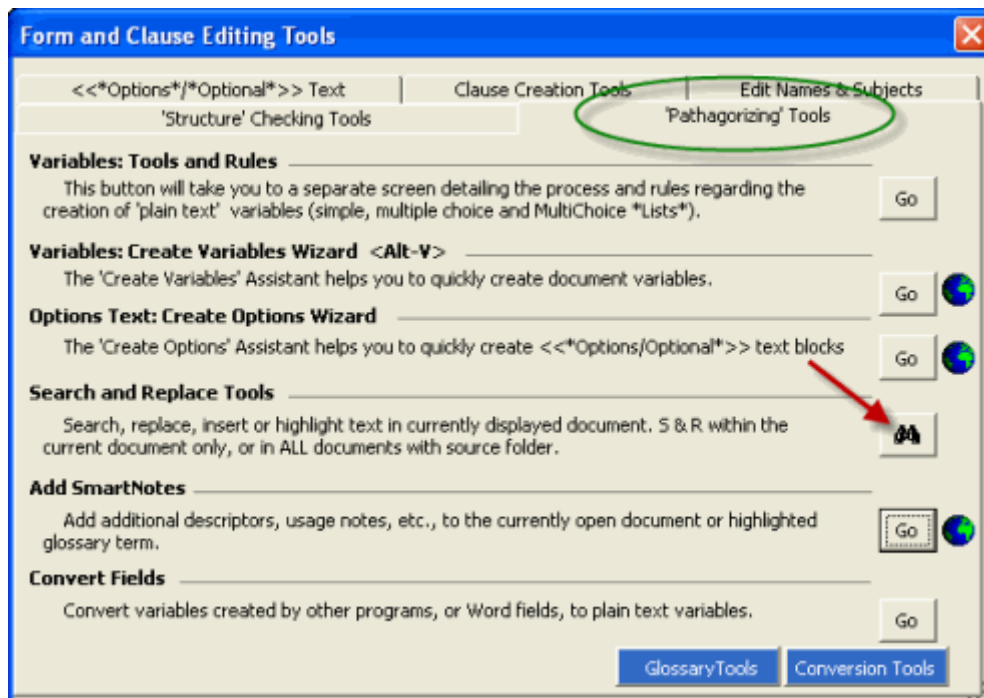
By selecting the appropriate options, your search and replace can be limited to the current document or expanded

- (a) to all documents in the current folder and
- (b) to all documents in child folders of the current folder.

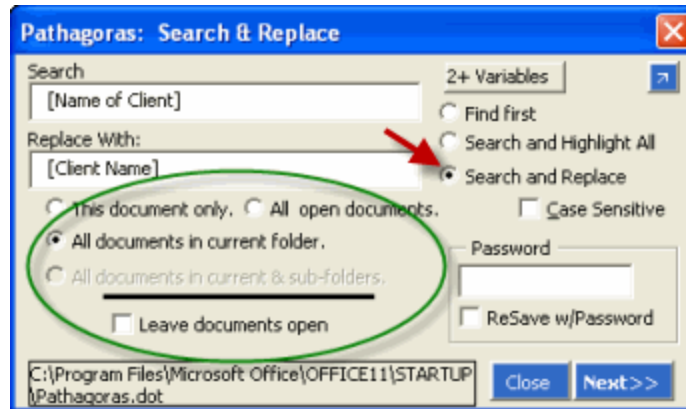
Activate Pathagoras' Search & Replace tool by clicking the 'Search & Replace' element in the Pathagoras features dropdown (found inside the 'Authoring/Editing Tools' sub-category



You can also activate Search and Replace from the 'Authoring/Editing Tools | General Editing Tools' screen



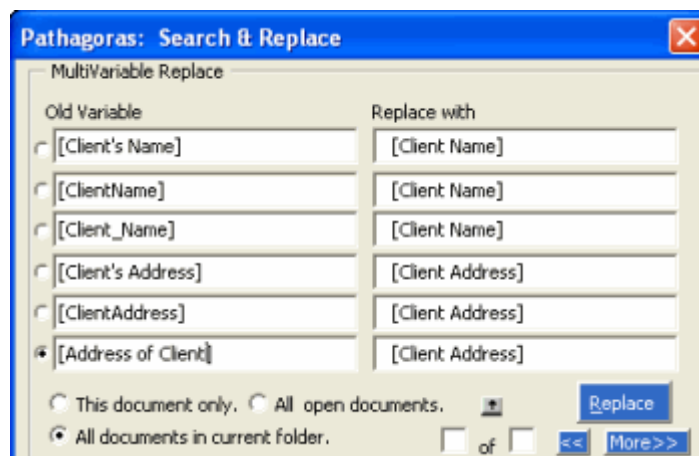
After the Search and Replace screen appears, you will be presented a variety of options as to action and scope.



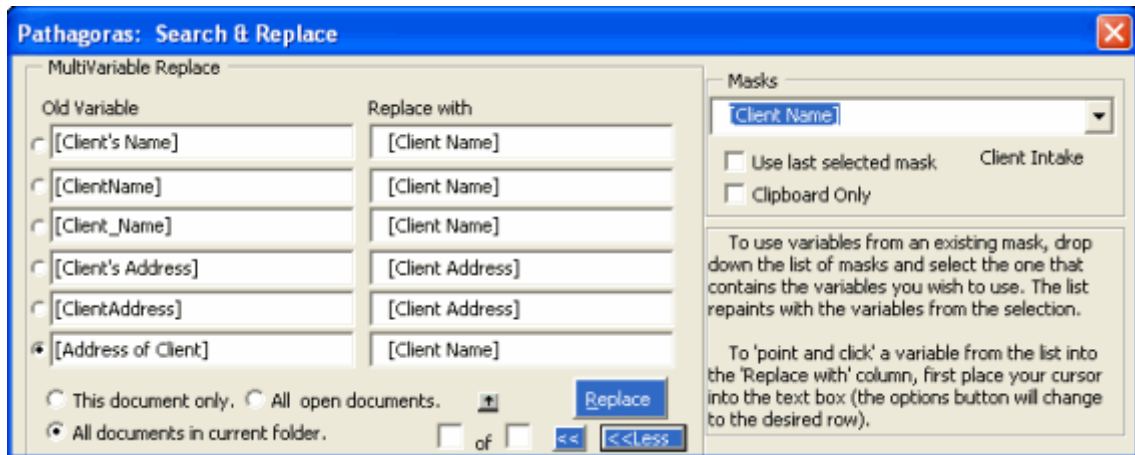
Note the 'action' and the 'scope' choices that are available to you with Pathagoras' 'Search and Replace' tool.

Multiple value Search and Replace

When you press the '2+ Variables' button, the screen will reset to one that will allow you to search for and replace multiple (up to 10) distinct values. This is useful for any type of search you want to conduct, but especially useful when you are trying to search for multiple variations of the same essential variable. E.g., early in your set ups, you used [Client Name] in some documents, [ClientName] in others and [Name of Client], or [Client's Name], etc., in still others. Use this tool to list the various permutations at the left and the appropriate replacement at the right. Before clicking 'Replace,' be sure to tell Pathagoras the scope of where you want replacement made



If you click the **More>>** button at the lower right, the screen will expand to show you all of your available masks (saved listings of variables) that you may have earlier created. You can select a mask and click in variables from that mask to insure consistency in naming your variables.



With the enlarged S&R screen, you can display your collection of masks. Select the desired mask and then select the variable with which you want to replace the current variables.

The next screen discusses how you can use the Instant Database as a Search & Replace tool similar to what is described above. Note, however, that the replacement action via the Instant Database screen applies only against current and open documents. The Search & Replace tool discussed on this page can make replacements folder wide.

You can activate the S&R tool while in the Instant Database screen. Just click the red 'Power Tools' button at the bottom of the IDB screen and click the 'Search & Replace' button.

Search Only: If you want to search for (but not replace) a term, place the search term in the box and click Next. If the term is not in the open document, Pathagoras will ask if you want to search for the term in other documents in the folder. If you say "Yes," Pathagoras will open each document in the folder until it finds the term. It will stop at the first instance it finds and highlight it.



Search & Replace is available as a *MyButton* in the Alt-Q (Quick Picks) menu.

23.4.2 Search and Replace via IDB

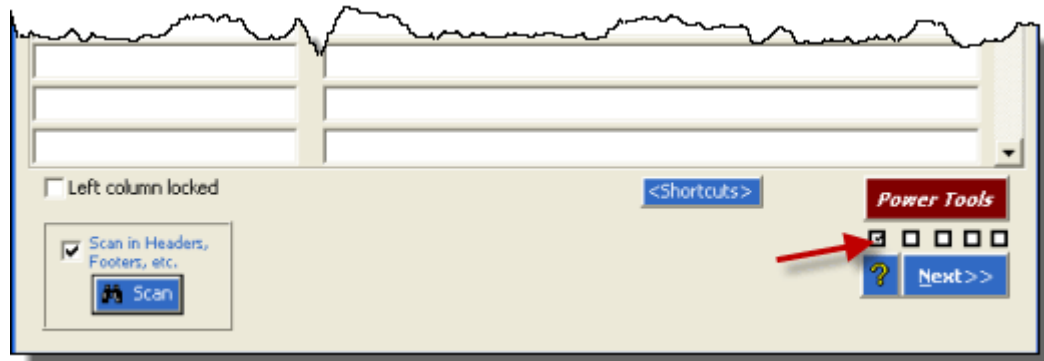
As stated [elsewhere](#)¹⁸⁶, the Instant Database module is, at its core, a search and replace device. Not only can it be used to replace document variables with personal values, it can be used to replace *any* text (not just variables) with any value.

If you have just a single word or phrase you want to change, you probably should just stick with Word's Find and Replace function. But if you want to perform searches and replaces on multiple words, consider the Instant Database.

Simply call up the Instant Database screen (Alt-D) and start typing. Type the 'old' word or phrase in the left side of the screen, and the replacement value at the right. To move from field to field, uncheck the "Left column locked" button in the lower left side of the screen. With IDB, you can replace a practically unlimited number of terms with new ones simultaneously. If you think you will might repeat the search and replace exercise at a later time, save the search and replace terms you have typed as a new record.

Bonus: when the first button in the Advanced Array (the series of checkboxes at the

bottom of the IDB screen) is checked, the replacements will be made against all open documents. So if you happen to have all (or at least several) documents open in separate windows, each of which contain the 'bad' text, consider using the Instant Database to perform a search and replace against all open documents.



The first checkbox in the Advanced Array tells Pathagoras to perform the replacements against all open documents.

Notes:

- Once you have created a list of 'search for' and 'replace with' terms on the Instant Database screen, ask yourself whether you might want to reuse this list. If you think you will, save it as a record or as a mask. Call it (for example) "S&R Terms" to distinguish it from client or customer records. By doing so, you can recall the record from the dropdown lists at a later time and use it over and over again.
- You cannot type an 'Enter' into the replace box. If you need to add a paragraph mark as part of the replacement text, use <P>.
- If you want to replace against *all documents in a folder*, use Pathagoras Search and Replace tool discussed in the previous section.)
- You can activate the S&R tool while in the Instant Database screen. Just click the red 'Power Tools' button at the bottom of the IDB screen and click the 'Search & Replace' button.



Instant Database is available as a **MyButton** in the Alt-Q (Quick Picks) menu.

23.5 Editing Tips

Pathagoras' primary document editing tools are discussed under separate sections of this manual. Check this table to see if what you want is here:

Structure Checker 173	Balance Markers 446
---	---------------------------------------

[Search and Replace Tools](#) 448[Wizards and Assistants](#) 454

Whether you are editing a source clause or a final document that you have assembled for a client or customer or patient, always keep in mind that you are always editing a Word document. All editing tools and features that Word has made available are always at your beck and call. You are never 'in Pathagoras' in the sense that your editing options are restricted.

So with this in mind, don't forget these tools:

The 'Undo' Tool:



(or Ctrl-Z)

Mistakes, typos, wrong clauses, etc. come with the territory of document assembly. With some programs when you find an error, it is back to the drawing board. Not so with Pathagoras.

If you have inserted a document from a DropDown List or from the Clause Selection screen, and decide you don't really want it you can manually locate and highlight the undesired text and delete it. But don't forget the Undo button (or Ctrl-Z, its keyboard equivalent). It may get you to where you want to be much more quickly.

If you used the Instant Database to make replacements in your document and see that you misspelled a value, press the Undo button (or 'Ctrl-Z') repeatedly until you get back to the 'original' state. (As you are repeating the Undo's, you will also see, in reverse, the actual steps used by the program to make replacements.)

There is a limit set by Word as to the number of steps you can undo. However, the limit is not fixed. You may be pleasantly surprised as to how far back you can go. (To see the number of 'Undo's' that you can perform, click the down arrow immediately to the right of the Undo button. The available Undo's are shown there.)

The Format Painter:



This is a 'cool tool' that allows you to capture a style or layout of a particular paragraph and apply it to any other paragraph(s) in the document. If you click the Format Painter once while in the paragraph that has the style you want to copy, you can make a change to another paragraph by mousing to the paragraph you want to change and clicking once. However, if you double-click the Format Painter, you can change an unlimited number of paragraphs. Just keep on clicking.

Testing Tips:

Test in Sections: If your document is long and 'sophisticated,' you may find it helpful to test just that portion of the document that is of immediate concern. That way you don't have to spend time answering questions regarding sections you know already work. Pathagoras plain text structure readily allows testing in sections. (Pathagoras does not require that the variables, optional text and other and other document elements remain available to the source.) Just highlight, copy and paste the section you want to test into a separate document.

Read more here about [Testing Your Documents](#) 160

Word tools: Don't forget that when you are editing a Pathagoras document, you are *always* in Word. Use regular Word functions when they make sense. For example, Word maintains

a Recently Used Files list. Use that list when you need to edit a file you just worked on. If you are editing, testing, reediting, retesting a document, this likely is the fastest way to access your work.

If you know that there is an error narrow down the scope of the 'balancing' routine by highlighting the block of text you wish to check before beginning the operation. This is especially useful in locating drafting errors in large documents with many markers and nested <<*Optional/Options*>> blocks.

BEFORE TESTING AN ORIGINAL DOCUMENT, MAKE SURE THAT YOU HAVE SAVED THE ORIGINAL OR THE IMPROVED VERSION OF THE DOCUMENT.

Processing the document changes it. The Options and Optional blocks will go away in favor of your choices. Repeats take away your carefully constructed commands and replaces them with the repeated text. With a good

AFTER TESTING ON AN ORIGINAL DOCUMENT, MAKE SURE THAT YOU DO NOT SAVE THE PROCESSED DOCUMENT. Ditto.

23.6 Wizards and Assistants


Pathagoras provides a number of Wizard and Assistant tools to make creating and editing documents easier and error free. Check these out:

Create Variables Wizard (Alt-V)	Create Options Wizard (Alt-O)
Paint Markers	Structure Checker
Search and Replace	Quick Picks

23.7 Bulk Editing Tools

There are several tools at your disposal that will enable you to make changes to either source text or assembled text in a global fashion.

But before going further, we want you to keep the following in mind. With regard to 'final' text, it likely is easier (or at least just as easy) to rebuild the document from scratch.

- The [Search & Replace](#)  tool discussed in a previous section is one -- good when you have a single change to make over a folder-full or documents.
- Where several words need to be changed, consider using the Instant Database. The 'first' box found in the 'Advance Array' cluster of boxes near the bottom of the Instant Database screen, if checked, will apply changes across all open documents. (Of course this requires that all documents in which you want to make changes be open.) Here are several possible applications.
 - ❖ If variables have not been assigned, then it is just a simple matter of assigning values to variables and applying the values to the variables in all open documents.
 - ❖ If variables have already been replaced AND the "Preserve Variables in Document" switch *was* active, you can simply recall the documents in which you want to make the changes. With any of the documents 'active', press Alt-D to activate the Instant Database. The 'Data Record' that was associated with the document will be displayed. (This happens automatically because the record was

attached to the document when it was originally personalized with data.) Simply change the variable to its new value. Before clicking Next, be sure to check the 'first' box in the Advanced Array' cluster to apply changes to all open documents. Click Next and everything will be updated.

- ❖ If variables have already been replaced AND the "Preserve variables in Document" switch was not active, you can use the Instant Database as a simple search & replace tool. This technique differs from Search & Replace in the first main bullet above in that you can change multiple values per document (up to 999, not that you would have that many) in one fell swoop.
- ❖ If there are many values to change, and those values already exist in an Instant Database 'Matter' record (albeit on the 'wrong' side of the form), keep in mind the 'reverse' function of the IDB. Here are the steps.
 1. Display the personal data record that you used to personalize the document to which you now wish to make changes.
 2. Click the red <Power Tools> button.
 3. Click the <Reverse Variable> to reverse the 'values' and variables column.
 4. Set the new values as appropriate. Delete lines containing values you don't wish to change.
 5. Note: In most cases, the above steps will allow you to restore variables to their original state, or to change values, but keep in mind that some 'reversals' will not be precise. For example, if the variable was [pronoun] and the value was "he" and you try to reverse that, any appearance of "he" (even one that you want to stay, or one that refers to a different actor, will be converted to '[pronoun]').

The Pathagoras System

Aliases and MultiChoice *Lists*

Part



XXIV

24 Aliases and MultiChoice *Lists*

An 'Alias' is a word or phrase that represents a list of multiple choice items. A 'MultiChoice *List*' is a series of multiple choices that has been assigned to a single word or phrase (the 'Alias').

*Think the 50 United States, or all the countries of the world,
or a product line, or the attorneys in your office or the counties served by your practice.*

The possibilities of what an Alias can represent is endless.

While only an Alias appears in your document, it is as if you embedded the entire MultiChoice *List* in the body of your document.

Before moving on to the details, let's start with a 'bad example' of presenting a long listing of **multiple choice variables** within your document.

The 'Bad' Example:

The following is a perfectly 'legal' but rather inefficient (not to mention confusing) way to create a multiple choice list which contains many elements (in the below example, 50 elements).

Travels 'R' Us is pleased to advise you that we have completed the itinerary for your summer vacation. On the first leg of your trip, we have scheduled you to visit the great state of [Alabama/Alaska/Arkansas/Arizona/California/Colorado/ Connecticut/Delaware/Florida/Georgia/Hawaii/Idaho/Illinois/Indiana/Iowa/Kentucky/Louisiana/Maine/Maryland/Massachusetts/Michigan/Minnesota/ Mississippi/Missouri/Montana/Nebraska/Nevada/New Hampshire/New Jersey/New Mexico/New York/North Carolina/North Dakota/Ohio/Oklahoma/Oregon/Pennsylvania/Rhode Island/South Carolina/South Dakota/Tennessee/Texas/Utah/Vermont/Virginia/Washington/West Virginia/ Wisconsin/Wyoming].

You will stay there for the first week and enjoy all the beauty this state has to offer. Then, you will travel to [Alabama/Alaska/Arkansas/Arizona/California/ Colorado/Connecticut/Delaware/Florida/Georgia/Hawaii/Idaho/Illinois/Indiana/ Iowa/Kentucky/Louisiana/Maine/Maryland/Massachusetts/Michigan/Minnesota/Mississippi/ Missouri/Montana/Nebraska/Nevada/New Hampshire/ New Jersey/New Mexico/New York/North Carolina/North Dakota/Ohio/Oklahoma/Oregon/Pennsylvania/Rhode Island/South Carolina/South Dakota/Tennessee/Texas/Utah/Vermont/Virginia/Washington/West Virginia/ Wisconsin/Wyoming] where you will enjoy the magnificent[vistas/ mountains/ beaches/ lakes/ monuments/ historical sites/ volcanoes/ gardens/ wide open spaces/ jungles/ canyons/ (other)] that this incredible state has to offer. We hope that you enjoy your trip.

As you can see, long multiple-choice lists within a document can get very distracting. It is simply too daunting for most end users to visually process.

A 'Good' Example:

The following is how the above could look:

Travels 'R' Us is pleased to advise you that we have completed the itinerary for your summer vacation. On the first leg of your trip, we have scheduled you to visit the great state of [*States*].


You will stay there for the first week and enjoy all the beauty this state has to offer. Then, you will travel to [*States*] where you will enjoy the magnificent [*sites*] that this incredible state has to offer. We hope that you enjoy your trip.

Much 'cleaner,' no?

Pathagoras lets you store a long list of variable choices (long being anything you consider as long) as a single word or short phrase which we call an 'Alias'. This 'alias' is placed in the source document in lieu of the long list. To tell Pathagoras that you are referencing a MultiChoice *List*, need only surround the alias by asterisks.

Here is an example of an Alias as a variable: "[*listname*]".

Here is the same Alias as simplified Options text: "{ *listname* }".

 The list of the 50 United States actually ships with the Demo and Retail versions of Pathagoras. The 'alias' is called "States." It can be referenced in the source document simply with "[*States*]" or "{ *States* }". Very simple. Very elegant.

Aliases 'in action':

When the alias is encountered during an Instant Database scan or during the processing of 'Options' text blocks, Pathagoras reads the multi-choices from the MultiChoice *List* table. The entire *List* of choices is presented to the end user for selection.

- (1) With **Instant Database**, the choices are displayed in a drop down list on the Instant Database screen.
- (2) With **Options blocks** (regular or simplified), the choices are shown in a selectable screen which displays during the actual document assembly process.

You can see the above [*states*] list in action by typing (or copying) the following sentence onto a blank document:

We will ship your widgets to [*States*] within [# of days to delivery] days of your order.

Then run Instant Database against the variables. (Reminder: to run IDB, press <Alt-D>. If the list does not automatically appear, press the Scan button when the IDB screen displays.)




A plea for assistance: Help to improve the world! Make a difference in the lives of others. Send me any 'universally usable' lists that you have created to:

lists@pathagoras.com.

I will post it on the website where it can be downloaded and used by others.

Conversely, you can download lists that I have already posted to the website that others have sent to me. Visit:

www.pathagoras.com/support/multichoicelists.html.

 Your MultiChoice *Lists* are stored in a file called "multichoice.txt". They are initially stored in the same folder that contains your Instant Database masks. [Click here](#) ⁴⁷⁸ to learn more about locating, sharing and re-locating the Multichoice *Lists* file.

See Also:

[Simple Multiple Choices](#) ¹¹⁸

[Cascading Options](#) ⁴⁶⁸

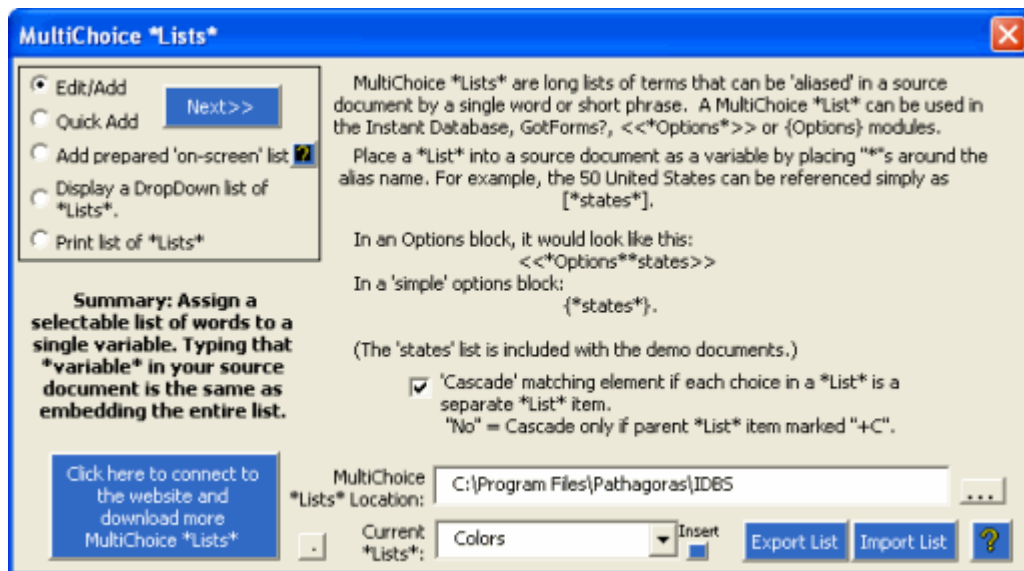
[Cascading Variables](#) ⁴⁷³

24.1 Creating Alias *Lists*

*Creating/Augmenting MultiChoice *Lists**

To assign, augment, edit, delete or otherwise modify a MultiChoice *List*:

1. Click the Pathagoras drop down features menu.
2. Click 'Editing Tools'.
3. Select the *Alias* element from the sub-list. You will see this screen:



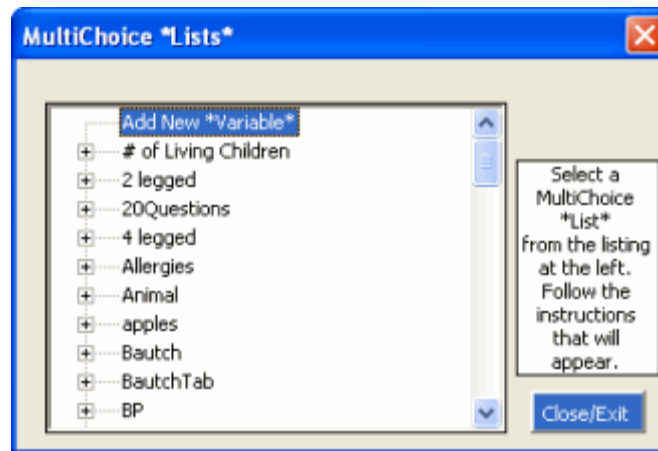
The Alias editing screen

(accessed through the entry titled Editing Tools|MultiChoice *Lists* in the Pathagoras Features Menu).

The following bullets describe the action of the three options presented in the upper left corner of the screen:

- The 'Edit/Add' option:

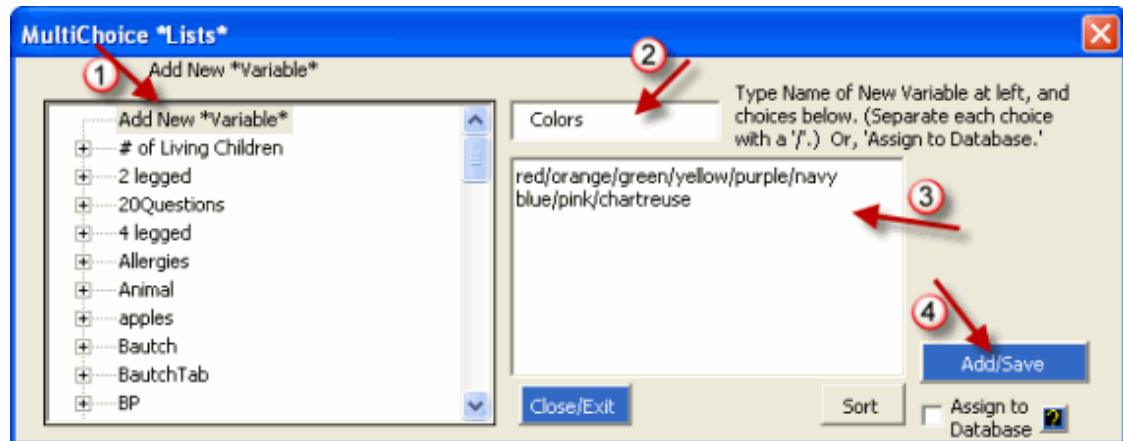
When you click Next>>, Pathagoras will take you to a screen like the following:



The List of Aliases

This table contains all of your current *Aliases*. Select the Alias you wish to edit.

To add a new variable and an associated list of choices, click Add New *Variable*. The screen will expand, providing you room to type the variable name and the list.



1. Click 'Add New *Variable*
2. Type name of new *variable*
3. Type list of choices, separating each with a '/' (slash).
4. Click Add/Save to complete action.

When you have finished adding new variables, close the screen.

- **The "Quick Add" option:** You can add a list (typically a short list, since you will be manually typing the entries) of multiple choices directly from the keyboard. When you select the Quick Add option and press Next>>, Pathagoras will ask you for the name you want to assign to the variable and then ask you to type up to 10 choices that you want the variable to represent. When done, click OK and Pathagoras will save the choices into your MultiChoice *List* collection.
- **The "Add prepared 'on-screen' list" option:** Pathagoras can add several kinds of 'prepared lists'.
 - Before starting, type a simple list of terms onto your editing screen. Separate each term by a slash (/).
 - E.g. red/green/purple/orange/navy blue/light blue/burnt orange/pink/chartreuse


After your list is prepared, highlight the collection of terms. (You have to tell Pathagoras what section of your on-screen document you want to add). Display the MultiChoice screen and select the "Add prepared 'on-screen' list" option. Press Next. Pathagoras will ask you for the alias you want to assign to the choices. It will then save the list in your MultiChoice *List* file.

- A table of terms, with the top row reflecting the aliases

shareholder	number of shares	price
Robert Jones	100	10
Samantha Peters	200	15.50
Andrea Munoz	200	16


- A table of terms, with the left-most column reflecting the aliases

shareholder	Robert Jones	Samantha Peters	Andrea Munoz
number of shares	100	200	200
price	10	15.50	16

- One of Pathagoras multi-dimentional Cascading Options Charts ([discussed elsewhere](#) , 472).

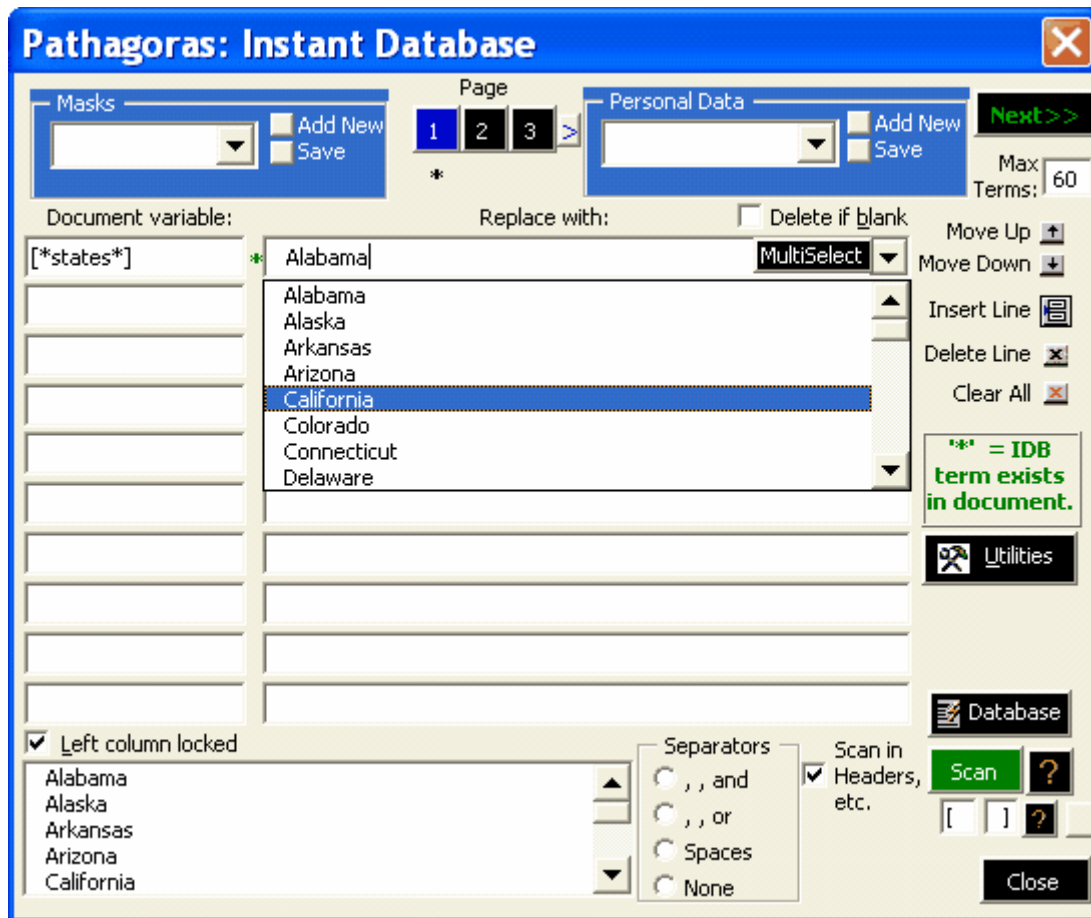
With regard to any of the above 'tables' options, make sure your cursor is inside the table before trying to 'Add prepared 'on-screen' list'.

Regardless of which of the above techniques you used to create your *Lists*, when you insert your *listname* into a document, the terms in your list will appear when you run an Instant Database or *GotForms?* scan, or process a document when the *List* is presented as an options block via: { *listname* } or <<*Options**listname*>>.

 **'Natural slashes':** Let's say your MultiChoice *List* text will contain 'natural slashes'. Perhaps you want to present the end users with a series of dates in the format 00/00/0000 from which to choose or an address contains an unavoidable 'slash'. You must use '/OR' as the separator between choices.(e.g., "01/28/1953/OR07/07/1988/OR6/22/99"). You must check the 'Use /OR' switch found in the Instant Database settings screen (Utilities/Settings|All Settings|Instant Database).

Visual Depictions of a *List* in action.

Just in case you haven't tried any of the above examples on your own computer, but you would like a quick peek at the [*states*] Multichoice *Lists*, here is a 'look-see' of what the results would be:



Instant Database display of the [*states*] variable list.

Note also the Multi-choice possibilities at the bottom of the screen.
You can select one, some or all of the list elements.
You can also indicate the separator, if any.

Here is the rendition of [*states*] in the *GotForms?* module:



Figure 4. *GotForms?* rendition of the Multichoice variable list "**states**".

Only the first five elements in the list will show on buttons. To show more, click the "Next 5>>" button. To insert a choice, just click on a button.

The use of MultiChoice *Lists* with Options is particularly useful when a fluctuating number

of variables needs to be brought into the document under construction. For example, in a Will, there might be several children, and a variable (e.g., [Name and Birthday of ChildX]) needs to be provided for each child. You could create a MultiChoice *List* called "children" and the values in the list might be:

[first child]/[first child] and [second child]/[first child], [second child]
and [third child]/[first child], [second child], [third child] and [fourth
child]/[first child], [second child], [third child], [fourth child] and [fifth
child]/[first child], [second child], [third child], [fourth child], [fifth child]
and [sixth child]

Here is how the above might render when { *children* } is encountered during document assembly. (The same result would obtain using <<*Options**children*>>).

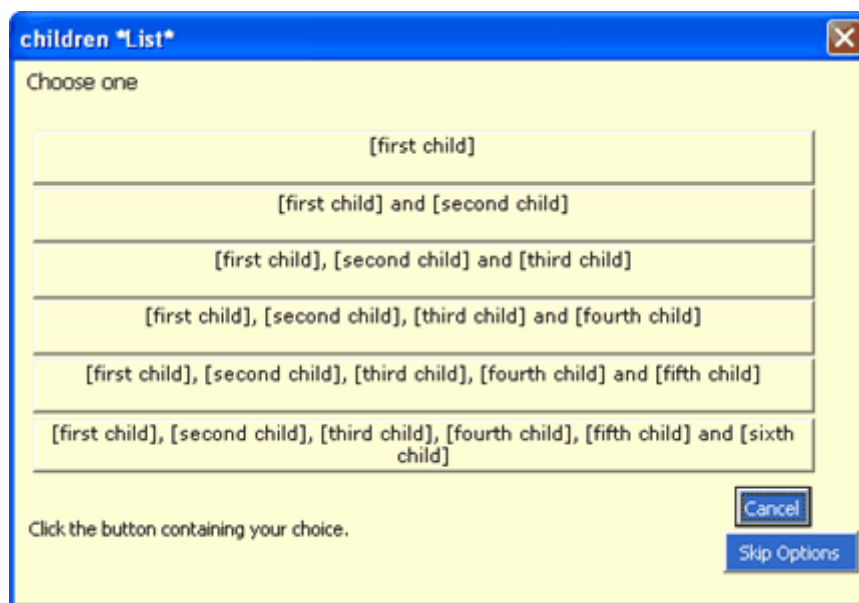


Figure 5. { *Children* } /<<*Options**Children*>> rendition in Pathagoras' options module.

Can I create Addresses (or other multi-line) MultiChoice items?

Yes. Pathagoras handles multi-line items without a problem. If you plan to use the MultiChoice list in an Instant Database screen (which does not allow true 'Enters', you must represent 'Enter' with "<P>". Pathagoras will 'decode' the <P>s at replacement time. See [<P>agraph Functions](#) ²⁴⁵

E.g.:

Big Store<P>123 Main Street<P>Anytown, PA 23454/Little Store<P>324 Oak Lane<P>My Town, VA 43234/etc.

Can I create a *List* from an existing external database?

Yes. See [this page](#) ⁴⁷⁹. This is a really 'cool' tool.



A final way to add a new *List* to your collection exists. Let's assume that you have a reference to a [*list* variable] in your source document. That *List* does not currently exist in your MultiChoice *List* collection. If you process the form with an Instant Database call, Pathagoras will notify you that the [*list* variable] does not exist and offer you the opportunity to create it on the fly. This is not a 'best practice' way to add to your collection, but it definitely works.

24.2 Easy As Pi *Alias* List

You can create a MultiChoice list from a variety of simple setups. Here are the main ones:

- From plain text: John Q. Adams/Thomas Jefferson/Abraham Lincoln/Richard M. Nixon/
Gerald R. Ford/William J. Clinton
 - Just separate the choices with slashes.
 - When you are ready to add your list to the MultiChoice system, highlight the block of text containing the list and then click on the MultiChoice item in Pathagoras features dropdown (Authoring/Editing tools). Click the ***'Insert Prepared List'*** from the choices. Then click Next. Pathagoras will ask what alias you want to assign to the list. Provide a one or two word alias. Pathagoras does the rest.
- From a column of text:
 - E.g.:

John Q. Adams
 Thomas Jefferson
 Abraham Lincoln
 Richard M. Nixon
 Gerald R. Ford
 William J. Clinton
 - When you are ready to add your list to the MultiChoice system, highlight the list. Click on the MultiChoice item in Pathagoras features dropdown (Authoring/Editing tools). Click the ***'Insert Prepared List'*** from the choices. Then click Next. Pathagoras will ask what alias you want to assign to the list. Provide a one or two word alias. Pathagoras does the rest.
- From a single column table
 - Type each entry into a separate row of the table

John Q. Adams
Thomas Jefferson
Abraham Lincoln
Richard M. Nixon
Gerald R. Ford
William J. Clinton

- When you are ready to add your list to the MultiChoice system, either highlight the table or at least put your cursor inside or just above the table. Click on the MultiChoice item in Pathagoras features dropdown (Authoring/Editing tools). Click the **'Insert Prepared List'** from the choices. Then click Next. Pathagoras will ask what alias you want to assign to the list. Provide a one or two word alias. Pathagoras does the rest.
- From a two column table.

Place the 'alias' you want to use in the left column and the MultiChoices (separated by slashes) at the right. Here is an example

Colors	red/blue/green/orange
Attorneys	John Q. Adams/Thomas Jefferson/Abraham Lincoln/Richard M. Nixon/Gerald R. Ford/William J. Clinton
Attorney Bar Numbers	1111/22222/33333/44444/5555
Attorney Telephones	(202) 555-1212/(205) 334-1452/(354) 555-3468/(890) 555-2009/(556) 444-4444
Sizes	X-Small/Small/Medium/Large/X-Large/XX-Large
Ice cream flavors	chocolate/vanilla/strawberry/Rocky Road/pistachio/cherry swirl/Neapolitan/banana nut
Mailing types	First class mail/First class (return receipt requested)/UPS/Parcel Post/FedEx/DHL/Facsimile transmission/E-mail/Carrier pigeon

- When you are ready to add your list to the MultiChoice system, highlight the table or place your cursor inside the table. Click on the MultiChoice item in Pathagoras features dropdown (Authoring/Editing tools). Click the **'Insert Prepared List'** from the choices. Then click Next. Pathagoras will create an entry for each alias and the associated choices.

24.3 Excel Alias *Lists*

In addition to the default storage of aliases within the Pathagoras system (stored in a file called "multichoice.csv" in the Instant Database folder), you can also maintain Alias *Lists* in Excel spreadsheets.

One type of spreadsheet is the kind that bears the name of the Alias itself. Two requirements for this kind:

1. The name of the spreadsheet must match the name of the Alias (in other words, if you want to call your Alias *List* "Attorney Name", the spreadsheet name must be "Attorney Name.xls" (or .xlsx)
2. The spreadsheet must be saved the folder in which your default Alias *List* file is saved. (You

default alias list is called 'multichoice.csv'. By default it is stored in the Instant Database folder. You can determine the current location of you Instant Database folder by pressing Alt-D (for Instant Database) and looking toward the bottom of the screen. The full path to IDB files is shown there in a text box. (You can copy the path to the IDB folder into your clipboard by clicking the clipboard image to the left of the text box.)

The other type of Alias storage by spreadsheet is within a file called "multichoice.xls" (or .xlsx). In this file, multiple Alias *Lists can be stored. (Note: the base file name is the same as the 'default storage' file. Only the extension is different.)

Here's how to set up this spreadsheet:

1. Create a new Excel spreadsheet and save it as 'multichoice.xls' or ('.xlsx') in your Instant Database files folder. You can determine the current location of you Instant Database folder by pressing Alt-D (for Instant Database) and looking toward the bottom of the screen. The full path to IDB files is shown there in a text box. You can copy the path to the IDB folder into your clipboard by clicking the clipboard image to the left of the text box.)
2. In row 1, column 1, name your first Alias (for example if you wanted to list the 50 United States, you might title it "States". You can use multiple words for titles, e.g., "United States of America", but generally shorter is better.
3. In row 2 of column 1, provide the first option.
4. In row 3 thru (infinity), type the text of all other options.
5. Repeat step 2 thru 4 for each new Alias, with the Alias always occupying row 1 and values beneath it. Use a new column for each alias. Do not skip columns. If Pathagoras sees a blank column, it assumes that you have no more aliases.

Using the *Lists*

Use the list as any other *Alias* list. As a variable, type it between two brackets (to indicate a variable) and between two asterisk (it's the asterisks that make it an Alias: [*Attorney*], [*States*], [*Bar Numbers*])

If you wish to group two of more Aliases together, just put a groupname inside of the variable brackets:

[!atty!*Attorney Name*] and [!atty!Attorney Bar Number]; [!st!*States*] and [[!st!*Capitals*]

24.4 Presenting Alias *Lists*

As the previous examples suggested, you can use MultiChoice *Lists* in a variety of ways. Just use the proper enclosure to signal Pathagoras what you intend.

MultiChoice *Lists* as Options text:

If you wish the *List* to be processed as the document is being assembled, present it as Options text. Here, the *List* is "states"

<<*Options**states*>>. (The doubling up of the asterisks is correct.)

For simple options, just use curly braces (don't forget the asterisks): { *states* }

MultiChoice *Lists* as Variables:

If you wish the *List* to be completed by the end user after the document has been assembled and as it is being personalized (i.e, when the user presses <Alt-D>, present it as a variable:

[*states*]

Use a MultiChoice *List* item in different locations to represent different values:

If the same variable Making unique names for each *List* is easy to do for a simple variable. Just append a number or other character at the end. E.g., "[*List*1]", "[*List*2]", "[*List*3]" etc.

Can I use !Groups! with MultiChoice *Lists*?

Absolutely. And this is a terrific way to pair up a name with an address and a phone number, or any other kind of association you can think of. Just precede each list that you want 'grouped' with a !groupname!. Make sure that (1) there is an identical number of options in each list and (2) that 'complementary' selections are in the same relative positions in each *List*. .

E.g.: [**!colors!*forecolors***] (and somewhere else in the document) [**!colors!*backcolors***].

We have provided a few pages below a more elaborate example of MultiChoice *List* and !Groups!. [Click here to go directly to that page.](#)⁴⁷⁶

Can I cascade MultiChoice *Lists*?

Yes. MultiChoice *Lists* can be nested/cascaded to 2 levels for Options, 1 level (currently) for variables. See [Cascading MultiChoice *Lists*](#)⁴⁶⁸.

Travels 'R' Us is pleased to advise you that we have completed the itinerary for your summer vacation. On the first leg of your trip, we have scheduled you to visit the great state of { *states*1 }.

You will stay there for the first week and enjoy all the beauty this state has to offer. Then, you will travel to { *states*2 } where you will enjoy the magnificent [*sites*] that this incredible state has to offer. We hope that you enjoy your trip.

24.5 Cascading Options

A 'classic' *List* provides the end user with options from which to choose. The *List* can comprise single words, short phrases or even links to other documents. But you can also tell Pathagoras to display another *List*, this second list being sub-categories of the selected parent. With this technique, you can allow a user to select from progressively 'narrower' sub-topics nested beneath the main topic. This progression from a broad topic to a narrow result is referred to as 'cascading'.

Let's say that you manage many real estate holdings of a variety of types. You want the selection of one 'property type' listing in a MultiChoice *List* to automatically present an appropriate sub-list from

which additional (but more limited) choices can be made. Here is what you might try:

First, create a MultiChoice *List* using the Cascading Options Chart. for the 'parent' choices. Continuing the above example, you might give the parent *List* a name like "Property Types" (typed in the left column of the MultiChoice *List* table), with the listing of the types of property in the right column.

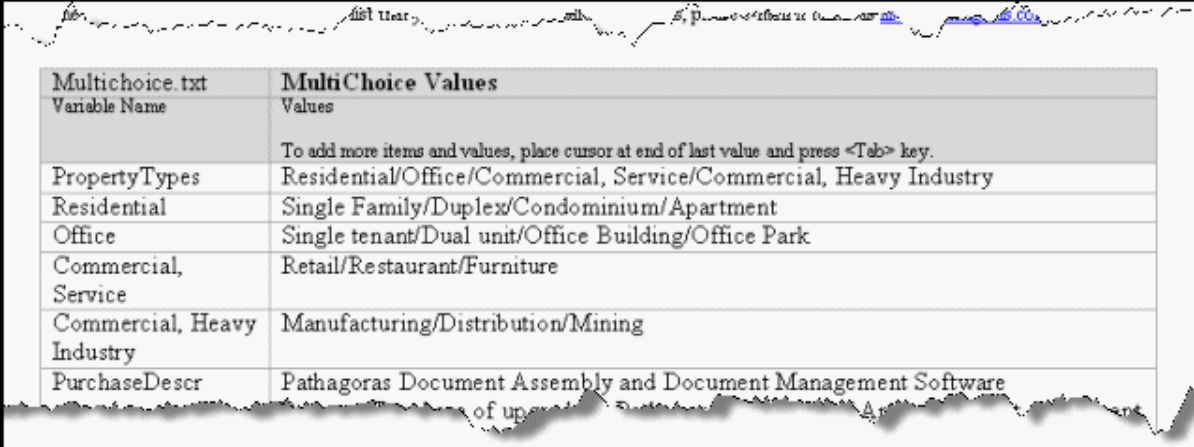
Property Type	Residential	Office	Commercial, Service	Commercial, Heavy Industry
	Single Family	Single tenant	Retail	Manufacturing
	Duplex	Dual unit	Restaurant	Distribution
	Condominium	Office Building	Furniture	Mining
	Apartment	Office Park		

Property Type	Residential/Office/Commercial, Service/Commercial, Heavy Industry
---------------	---

Next, create a separate MultiChoice *List* for each 'Property Type'. This is how the rest of the MultiChoice *Lists* might appear:

Residential	Single Family/Duplex/Condominium/Apartment
Office	Single tenant/Dual unit/Office Building/Office Park
Commercial, Service	Retail/Restaurant/Furniture
Commercial, Heavy Industry	Manufacturing/Distribution/Mining

So here is how the above *Lists* might appear in the MultiChoice *Lists* editor. (Keep in mind that everything is plain text in the MultiChoice *Lists* editor. The editor is a simple Word document with a table that contains the list name that you choose at the left and the assigned values that you create at the right.):



Multichoice.txt Variable Name	MultiChoice Values Values
	To add more items and values, place cursor at end of last value and press <Tab> key.
PropertyTypes	Residential/Office/Commercial, Service/Commercial, Heavy Industry
Residential	Single Family/Duplex/Condominium/Apartment
Office	Single tenant/Dual unit/Office Building/Office Park
Commercial, Service	Retail/Restaurant/Furniture
Commercial, Heavy Industry	Manufacturing/Distribution/Mining
PurchaseDescr	Pathagoras Document Assembly and Document Management Software

Note: when you are adding a 'child', you are actually creating another list within the MultiChoice *List*

collection at the same level of the parent. You are not actually creating a sub-list. Pathagoras automatically presumes nesting when it sees that each choice in a parent list exists as a separate *List* elsewhere in the table of terms. The benefit of this approach is that you can use **any** list at the parent level, even one originally conceived as a child.

You can nest to an unlimited number of levels. Simply create a new MultiChoice *List* for each level. Keep the above 'Note' in mind.

Now on to the actual document assembly steps:

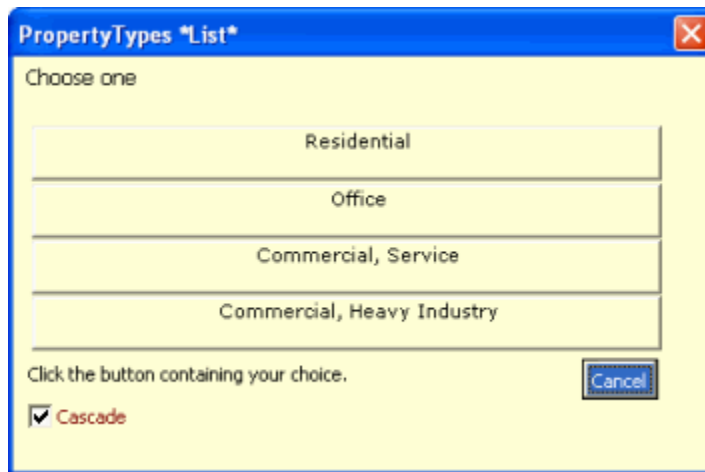
1. At a strategic location in your source document, type the top most level of the cascade into an options block.

- **Note:** You can use the 'standard' options block style: <<*Options**PropertyTypes*>>. (The double asterisks in the center are correct. The first asterisk closes the administrative section of the block. The second is the opening boundary for the *list* name.).

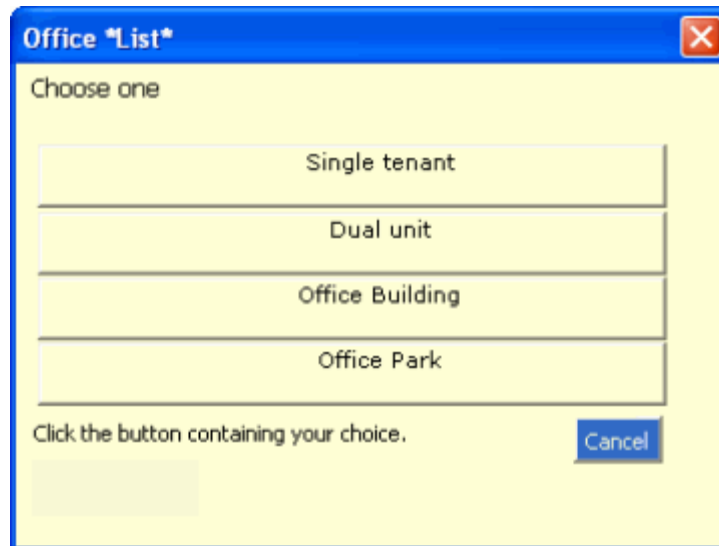
Alternatively, and probably easier, you can use the 'simplified options' block layout using {curly braces}. Example: {*PropertyTypes*}.

2. Assemble a document which contains the *List*. When Pathagoras encounters the *List* reference, it will detect the existence of 'cascading'. Pathagoras redraws the options screen, presenting the next level of choices).

Here is what the initial options screen would look like in the above example when {*PropertyTypes*} is processed. Make sure the Cascade button is checked. (It will be displayed and automatically checked when Pathagoras detects that a cascading possibility exists.)



When the Cascade button is checked, and if you click "Office", Pathagoras will present the following choices:



Note that the Cascade button has 'disappeared.'

That is because Pathagoras has determined that none of the current choices are themselves *Lists*. No further cascades, therefore, are possible.

i If, in the initial screen, you unchecked the Cascade button, the *text* of the selected choice would have been inserted in place of the MultiChoice *keyword*. This of course is the default action of a selection from an Options screen, and sometimes may be the result that you seek.



There are no practical restrictions as to what the 'reference' contained in your choices can be (or point). The choices can be another level of choices which allow for further cascading, or they can be the 'final choices' or lowest level. The lowest level of nested choices can likewise be anything:

- It can be just plain text. The examples provided above would insert the plain text 'answer' into the document in place of the original *List* term.
- It can be a [Variable]. Just enclose the lowest level choices within "[" and "]" markers. After the document is assembled and when you call up the Instant Database via <Alt-D>, you will be presented the variable for completion with a personal value.
- It can be a MultiChoice [*Variable*]. Just enclose the lowest level choices within "[" and "]" markers. After the document is assembled and after you call up the Instant Database via <Alt-D>, all choices within the *List* will be displayed in dropdown list fashion alongside of the variable name.
- It can be a 'pointer' to a document. To reference a document, just surround the document name with "<<" and ">>" markers. The document reference can be 'complete' with drive and path designators (e.g., "C:\my documents\Letter to Clerk.doc"), or if you are using the prefix suffix designators for your document, or one of the [SuperBooks](#)^[90] that Pathagoras allows, you can just type the term's short name. Pathagoras will locate these latter documents using standard [order of search rules](#)^[95]. (We very much want you to click this link. Understanding the 'order of search' rules will allow you create some tremendous document assembly possibilities.) So, in the above example, instead of the list following Residential being this in the above example:

Residential	Single Family/Duplex/Condominium/Apartment
-------------	--

you could provide this:

Residential	<<res100>>/<<res200>>/<<res300>>/ <<res400>>
-------------	---

The assumption made here is that res100, res200, etc. are documents. (The further assumption is that "res" is a [prefix](#)⁴²⁰ that has been registered to point to the folder that contains the desired clauses, or that the documents reside in a SuperBook.)

One last point. In the last example above, the lowest level of the cascade are individual documents or clauses. But imagine that the lowest level pointed to clause-sets. A [clause-set](#)⁴⁰⁶ is a 'potential document' that can range from simple to one of extra-ordinary complexity. You could use the cascading tools described above to ask progressively more focused questions which, when answered, lead to the production of a highly specialized, perfectly tailored and complete document.

24.6 Cascading Options Chart

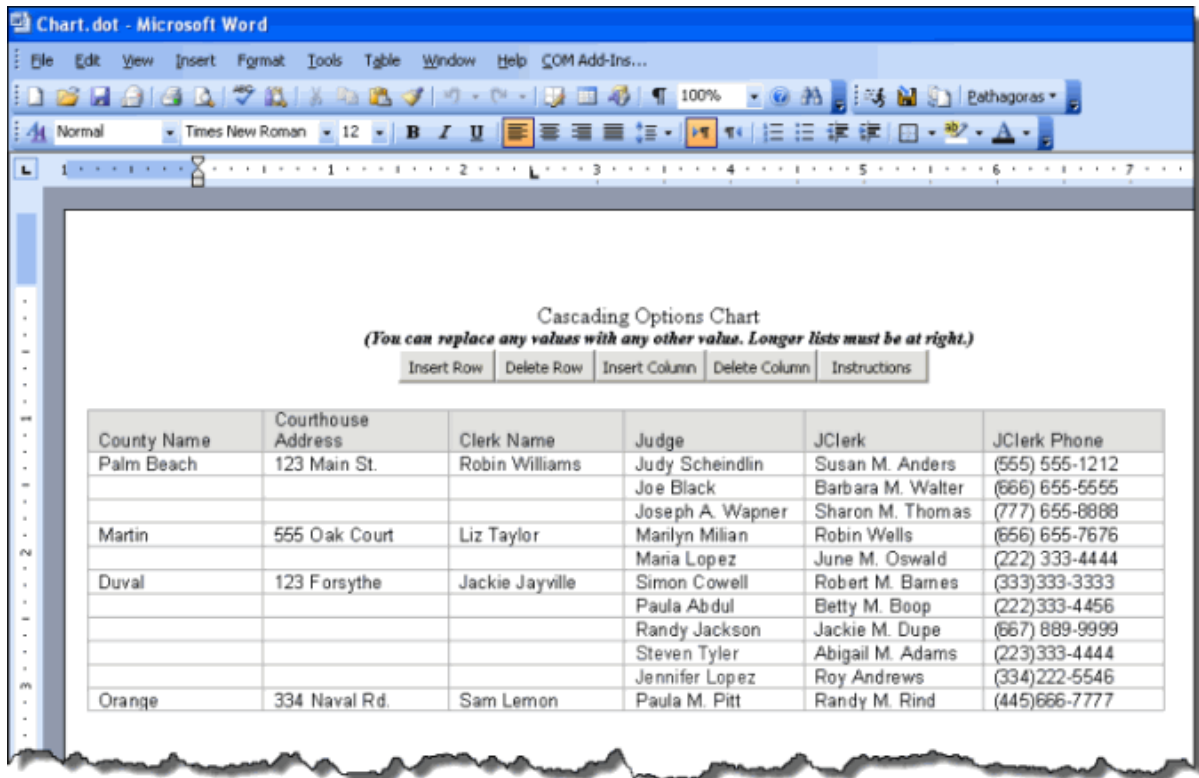
Setting up the various cascading values can be tedious if done by hand. Lining up the variables, insuring the proper count and proper spellings presents not insurmountable challenges if done by hand, but difficult ones nevertheless.

Pathagoras to the rescue with its Cascading Chart. This is a document that ships with Pathagoras that provides you an easy to use template which you complete with the various elements.

To use, navigate to "C:\Program Files\Pathagoras\Chart.dot". (If yours is a 64 bit machine, the path is C:\Program Files (x86)\Pathagoras\Chart.dot".)

Double click on 'Chart.dot' to open. Depending upon your security settings, you may be asked to confirm running the macros found in the chart. (The macros control the buttons in the Chart which allow you to display the instructions and to add and delete rows and columns as you are adding elements.)

Here is what you should initially see:



Using the values you want in your MultiChoice *Lists*, complete the chart as appropriate. when done, you can automatically add the contents of the chart to the currently stored *Lists* by:

1. Click Pathagoras Features dropdown
2. Select Authoring/Editing tools and
3. Select MultiChoice*Lists*;
4. From the resulting screen, select 'Import Prepared List' and click the Next button.
5. Pathagoras will quickly import the data, creating the proper element for each *List* and report its success to you a confirmation screen.

24.7 Cascading Variables

When you select one item from a group of choices, it is often desirable that resulting display be another group of choices which are narrower in scope and intimately related to the first choice. This progression from a broad topic to a narrow selection is referred to as 'cascading'.

An Instant Database variable can be set up to cascade into other variables. It is an advanced program feature. We advise that 'cascading' should be undertaken only if the user is fully familiar with the (1) Instant Database, (2) the creation and use of multiple-choice variables, (3) the creation and use of MultiChoice *Lists*, and (4) the use of !groups! to 'connect' two or more variables together.

Assuming that familiarity, however, the setup of cascading lists can be remarkably quite simple. ("Cascading MultiChoice *Lists* It is a cousin to "Cascading Options" discussed in

the previous section. While the cascading principles are quite similar, they are not identical.

Examples:

Top Level: "County". This could be a MultiChoice *List* variable that contains let's say the counties in which you conduct your practice or trade. We'll use these for purposes of the example:

"Baker/Nassau/Duval/Clay/St. Johns/Putnam/Flagler"
(these are the counties in Northeast Florida)

Connector Level: "Judges": This variable would contain references to a series of MultiChoice aliases, each one of which contains the names of the judges who sit in the individual jurisdictions. The actual choices need to reference other lists, but the connector level is not actually displayed. To continue with the example, the connector list called "Judges" might read "Baker Judges/Nassau Judges/Duval Judges/Clay Judges/St. Johns Judges/Putnam Judges/Flagler Judges".

Lists Level: The names of the actual judges are provided in what might be called a sub-sub lists. Each of these lists would match the names found in the Connector lists. Continuing the example, there would be one list called "Baker Judges" containing, let's say, "Judge John Jones/ Judge Ronald Adams/Judge Frank Smith"; a second list called "Nassau Judges" would containing the names of all of the Nassau County judges); a third list called Duval Judges, etc.

The document setup:

Your document would contain at least two variables. [*County*] and [*County*:*Judges*]. When you select a specific county at the topmost level, the selection at the top will cause the judge list for the selected jurisdiction to display.

Requirements: Not many.

1. You must have a 'top level' MultiChoice *List*. This list will reference other MultiChoice *Lists*.
2. You must have second level *Lists* that will pair up with the choices in the top level list.

As will be explained below, this pairing up is by position, not by name. Therefore, you are not restricted in how you might name either the top level lists or the secondary lists.

3. You join the top list and any secondary lists together via a !groupname!. It is the !groupname! that signals them to change in tandem.

There is nothing special about the setup of the 'top level' list. Just type the names of all of the second-level choices you want to present to the end user. As is typical in creating a MultiChoice List, separate each choice with a slash.

Note: It is not necessary that each second-level choice in fact exist (by name) in your collection of MultiChoice Lists. You must have, however, have the same number of separate elements at the connector level as you have choices in the top level. The way the initial 'cascade down' works is by position, not by name. That way, the choice of County can also be used to select other lists and sub-lists. The actual-list (which is only indirectly accessed -- by position not by name) must contain 'true' names of lists.

So, following the above example, if you have 15 Counties, you must have 15 separate "XXXX Judges" (Baker Judges, Duval Judges, etc.)

(Everything being plain text, you can type up your top-level *List* even though the second-level lists don't yet exist. However, before trying Cascading, you must in fact create all second-level Lists shown in the top level.

Those 'connector level' MultiChoice *Lists* names must precisely match the names in the top-level lists. Each secondary list item contain the elements you want to cascade down to, with each choice separated by a slash.

Other examples:

Top Level

"Bank Name", "Charter Bank/America's Bank/Friendly Bank/Fort Post Credit Union"

Intermediate Level

"Bank Contact", "CBCon/ABCon/FBCon/FPCUCon"

List Level

"CBCon", "Mr. Jeffrey A. Biegel"

"MICon", "Ms. Rosanne Klarr"/Mr. Jackson Andrews/Ms. Stepanie Jacskon

"RVCon", "Mr. Peter Paulson/Ms. Clara Nowak"

"RCUCon", "Mr. Kevin Gardner/Mr. Alfred Honeywell/Mr. Frederick McCollum"

You can have any number of other lists that link to the top level via a !groupname!. The choice from the other lists will be made when the top level choice is made. Example:

"Bank Address", "1010 West Golfair Avenue<P>Newport News, VA 23602/411 North Bridge Street<P>Hampton, VA 23666/327 North Avenue<P>P.O. Box 777<P>Williamsburg, VA 23185/200 Riverfront Road<P>Norfolk, VA 23898"

When setting this up in your documents, your variables might look like this. (Note the groupname !bank! ties all of the variables together.):

[!bank!*Bank Name*]

ATTN: [!bank!*Bank Name*;!contact!*Bank Contact*]

[!bank!*Bank Address*]

(body of letter)

Note some aspects of the above.

As noted a few paragraphs above, "Bank Address" is not cascaded. It is at the same level as Bank Name. When you choose a Bank Name, the Bank Address will be selected based on the location of the item selected in the 'Bank Name' list so long as a group name is provided to tie the two together.

The cascading links are set out after a colon (which mere indicates the cascade. It also requires a 'groupname'.

=====

We provide an alternative approach to cascading which more 'intimately' links the top level to the secondary levels. Instead of linking by !groupname!, you can link the lists in the following perhaps more direct fashion.

Requirement: The two fields must appear on the same page of the Instant Database screen and must be adjacent to each other.

Setup: The first element must reflect the top-level. In the above example, [*PropertyTypes*] would appear exactly in that fashion. The next level is the top level with the addition of the word "Sub". e.g., [*PropertyTypes*sub]. (The 'sub' means that it is a sub-category of 'PropertyTypes'.

When the appropriate value is selected from the list shown in the Instant Database screen (and continuing with the above example, it will be from the *PropertyTypes* list, i.e., "Residential or Office or Commercial, Service or Commercial, Heavy Industry", the appropriate sub-list (in this case, we selected 'Residential' will show in the immediately following line on the Instant Database screen, like this:

Select item from parent list (1) and the sub-list (2) is populated with the proper values.
Drop down the list, point and click.

24.8 Aliases and !Groups!

!Groups!

The !Group! function works with MultiChoice *Lists* and Options. (As explained [elsewhere](#)¹¹⁹, a !Group! allows a user to select an element from one member of the group. It results in the automatic selection of the proper entry from other members of the group.)

Pairing MultiChoice *Lists* with Pathagoras' !Groups! feature is a perfect marriage, one practically unlimited in its potential. When you select one item from a particular *List*, all parallel items from related *Lists* (related via the !Group! assignment) are automatically selected.

Consider the following. Type (or copy and paste) the following into a document:

The capital of [!St!*States*] is [!St!*Capitals*].

Press Alt-D to call up the Instant Database. Scan the document if the variables don't appear. Select a state (or a capital) from the drop down lists at the right of the Instant Database screen. You should see how the grouping enables the selection of one item to automatically reset the other.

OTHER EXAMPLES:

- **In a law office setting:** You have a list of cities in which you practice. You also have a list of Clerks, and a separate listing of their addresses, with whom you file pleadings. You also have a list of Sheriffs from whom you request service of process. Let's further assume that you have created individual MultiChoice *Lists* for each element (*city*, *clerk*, *sheriff*, *clerkaddress*, *sheriffaddress*, etc.)
- **In a non-law office setting:** You have a list of corporations with whom you conduct business, and a list of contact points within each business to whom you send the request for proposals, etc. Let's further assume that you have created individual MultiChoice *Lists* for each element (*customer*, *contact person*, *contactaddress*, etc.)

Your MultiChoice *List* table that ships with Pathagoras actually contains the above entries (note, however, the entries in the sample text actually starts with the prefix "Sample". We did so in order to allow you to use the actual variable names 'city', 'address', etc., in your 'real' data, if you choose:

Variable Name	MultiChoice Values for GotForms? and InstantDataBase
	To add more items and values, place cursor at end of last value and press <Tab> key.
month	January/February/March/April/May/June/July/August/September/October/November/December
Practice Areas	Domestic/Criminal/Estate Planning/Intellectual Property/Bankruptcy
SampleCity	Redding/Hampton/Jacksonville/Yorktown/Williamsburg
SampleClerk	Mary Rowe, Clerk/Sally Howe, Clerk/Robert James, Clerk/Clarence Yancey, Clerk/Andea Williams, Clerk
SampleAddress	123 Richmond Road<P>Redding, Arkansas 73432/1123 Hampton Highway<P>Hampton, VA 23666/344 Jackson Blvd. <P>Jacksonville, FL 32211/221 York Road<P>Yorktown, VA 23690/5445 Walls Lane<P>Williamsburg, VA 23185
SampleContacts	John Doe/Jane Rain/Mary Perry/Fred Hempshead/Alan Allen

Let's now assume that you have created a form document that contains these variables. Note that each variable that you want to change 'in tandem' is preceded by a !Group! name:

Jurisdiction: [!City!*SampleCity*]

[Date of Letter]

[!city!*SampleClerk*]

[!city!*SampleClerkAddress*]

Re: [Client Name] vs. [Defendant Name]

Case Number: [Case Number]


Please file the enclosed documents among the other papers:

[List of Documents]

Thank you for your kind attention to this matter.

Sincerely,


[AttorneySignatureBlock]

 In the above example, the !groupname! we chose to link the various variables is 'City!'. Please understand that the term you select for a !groupname! is not important. It could have been a non-sense word like !grzb! or a number like !123!. The !groupname! is in no way related to the variables it modifies. It is only an identifier used to tie the 'grouped' variables together.


When you run the Instant Database routine against the above text, all bracketed variables will appear in the left side of the Instant Database screen. The potential values for each variable will be displayed in a dropdown list at the right. Select any one of the !city! variables and the parallel selection of the others in the group will be automatically selected for you.



By the way -- you can easily duplicate all of the steps in the example we provide above. (Depending upon when you downloaded the program, you should already have the 'sample...' lists in your MultiChoices *List* table.) Simply copy the above sample text and paste it into a new document in Word. Press <Alt-D> to run the Instant Database and press the <Scan> button. Select any choice from any of the multiple choice dropdown lists.


 You can use MultiChoice *Lists* not only with variables (as shown above) but with robust <<*Options*>> and {Simplified Options} blocks as well. Here is the structure:

Robust Options: <<*Options*!GroupName!**Alias*>>

 The double asterisks between the GroupName and the Alias is mandatory. The first of the doubled asterisks closes the administrative section of the Options 'call to action'. The second * identifies the MultiChoice alias. The double asterisks are not required in the below two examples.

Simplified Options: {!GroupName!*Alias*}

Variables: [!GroupName!*Alias*]

See Also: [Groups](#)  119

24.9 Alias Files Location

The physical location of your MultiChoice *Lists* is initially in the same folder as your Instant Database records. (This location is initially "C:\Program Files\Pathagoras\IDBs".) However, you may, for any reason, change this location. If you do, you must 'repoint' to its location.

Your goal in this exercise could be for one of 3 purposes:

- (1) just to see where the MultiChoice *List* file is currently located;
- (2) to move the physical storage location of the current records; or \
- (3) to reset the pointer to the location where the MultiChoice *List* you really want to use is currently stored.

You can determine (and then reset) the current 'pointer value' for your MultiChoice *Lists* in two ways.

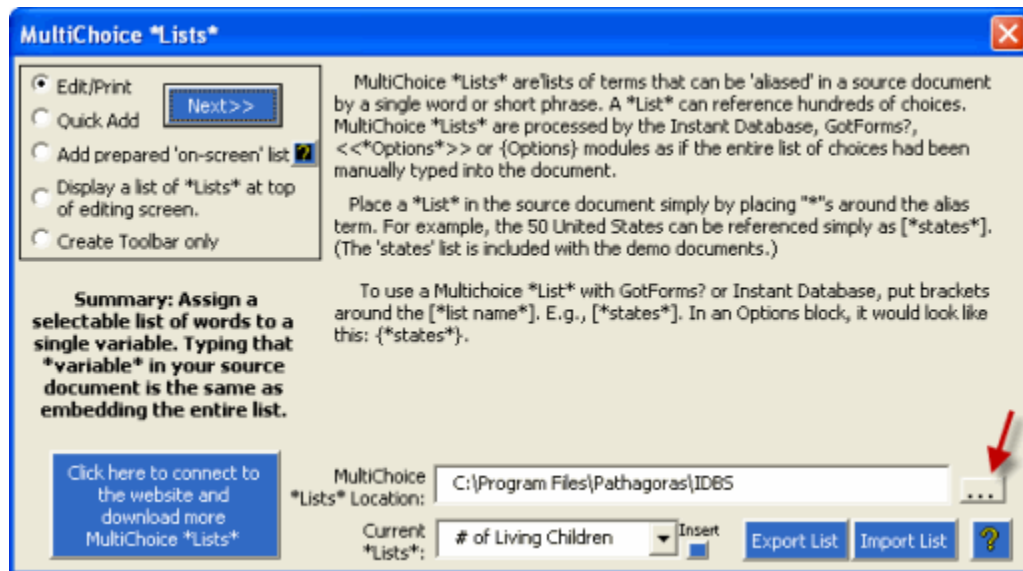
Via the Pathagoras drop down features list:

Click Editing Tools from the list and then the MultiChoice *Lists* entry from the sub-menu.

From the Utilities/Settings screen:

Click the <Show All Settings> button from the face of the Utilities/Settings screen and then click the MultiChoice *Lists* tab.

To move the files, or reset the pointer (and optionally move the content of your MultiChoice *List*), click the button bearing the ". . ." and follow the instructions. Navigate to the desired location and lock in the new location by pressing "OK".




When you have finished repointing, and if a MultiChoice *List* exists at both the original and the target locations, Pathagoras will ask whether you want to combine the two lists (and if so, how duplicate entries should be handled) or keep the original or the target *List*. Respond as appropriate.

24.10 Aliases from Database

Using the MultiChoice *List* module, and simple 'plain-text' references, Pathagoras can connect to data stored in most of your external databases. Setup is quick and intuitive. It involves simply pointing to the database, selecting a field, and providing a MultiChoice *List* name that will be used in your source documents. Thereafter, a call to the database from the source document

can be done with simple, plain text reference to that *List* name.

=====
 Currently, the *Lists* using databases feature works with Outlook, Excel (.xls*), and any comma delimited (.csv) and tab delimited (.txt or .tsv) files.

Such is not necessarily a serious restriction. These files encompass the universal data storage possibilities.

Most Pathagoras users already use Outlook and Excel.

Further, almost all databases can create Excel, comma or tab delimited files.

=====
We don't yet know if there is a maximum number of elements that Pathagoras can process, but in testing it working exceptionally fast on multiple databases with 2,100 entries each.

The previous pages demonstrated how to add a new, or edit an existing, *List* of terms in the MultiChoice *List* table. They involve manually typing (or copying from separately prepared list) the choices into your *List*. This contemplates that the *List* will be more or less static.

This works well for most circumstances. The 50 United States is static. The counties in which you conduct your practice is probably fairly static. If not static, it likely short enough to easily modify using the standard MultiChoice *Lists* editing tools described in previous pages.

In many circumstances, a hand typed, fixed list should be entirely satisfactory.

But what if choices that you want displayed is really long? Hundreds (thousands) of entries. What if the values frequently change? New names of customers are added and old ones deleted. Clients move to new addresses. New product models are added, others are retired.

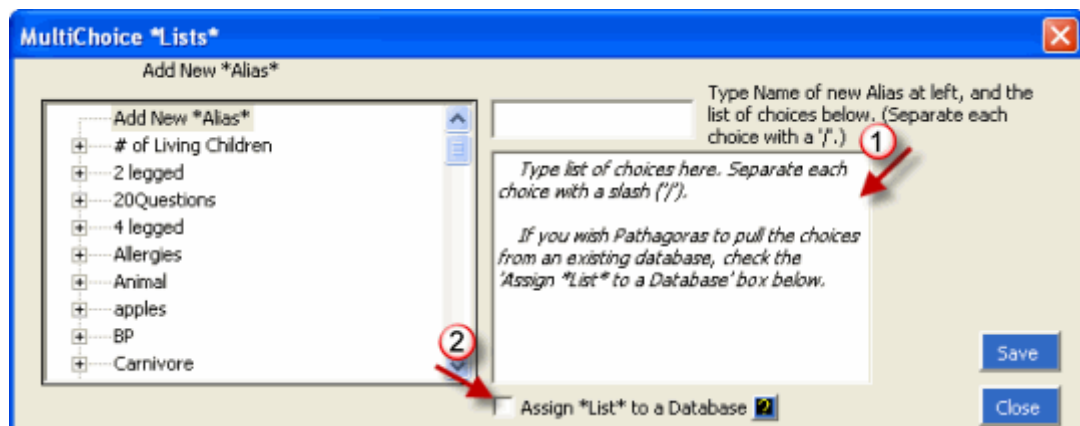
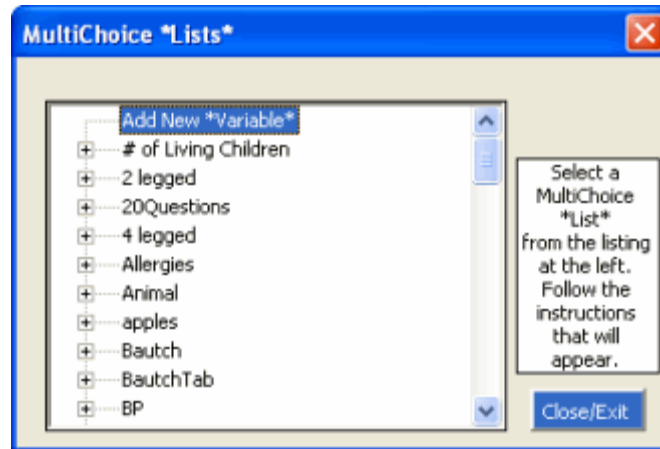
This is the 'stuff' of databases.

And Pathagoras can link to them. (We recognize that databases can be static too. If you have a *database* containing the 50 United States of America, it doesn't change. Just recognize that you can link to a static database just as readily as you can to a dynamic database. The message we are trying to convey here is that Pathagoras can link to any kind of database.)

You can easily populate a MultiChoice *List* from a database. The steps are easy. Just like just about everything else in Pathagoras, you simply 'point' to it. You also need to assign the database field to an 'alias' and make sure that the source document in which you intend to call in the list of choices to complete the variable contains the alias.

Here are the steps to create an alias that links to a database:

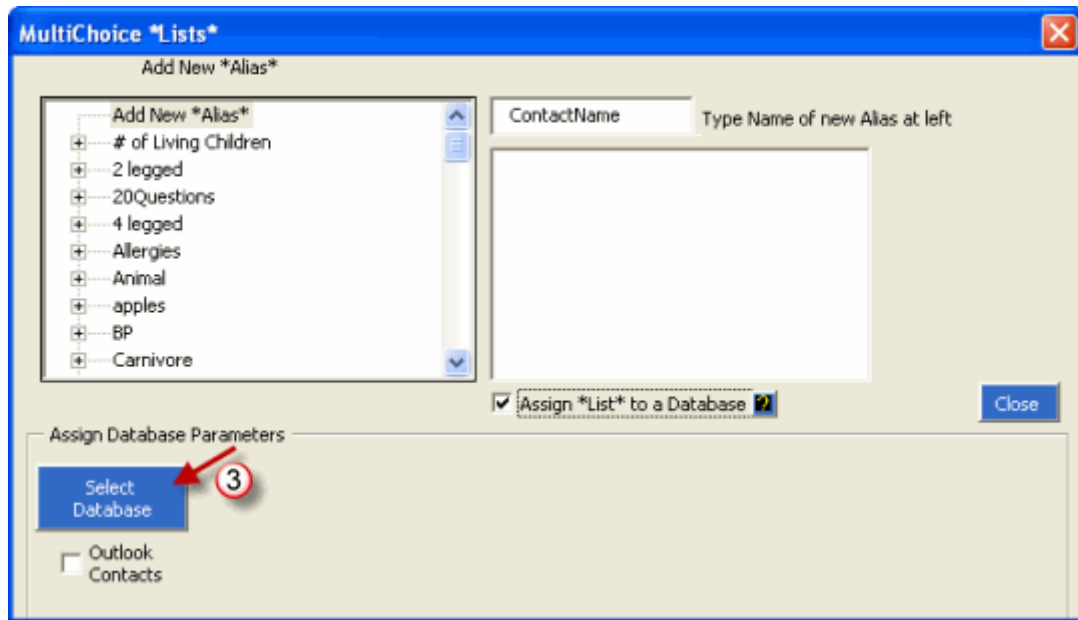
1. Display the MultiChoice *Lists* screen.
2. Click the Next>> button indicating that you want to Show/Edit terms. You will see this screen. If you want to add a new variable, click the top element. If you want to edit an existing entry, click that element.



We are presuming here that you want to add a new variable and that you want to connect it to a database.

- (1) Read instructions. Follow those in the second paragraph and Check Assign to Database
- (2). This expands the screen to reveal the Select Database button. (You can provide the alias for the new *List* at anytime.)

The following expanded screen will appear:

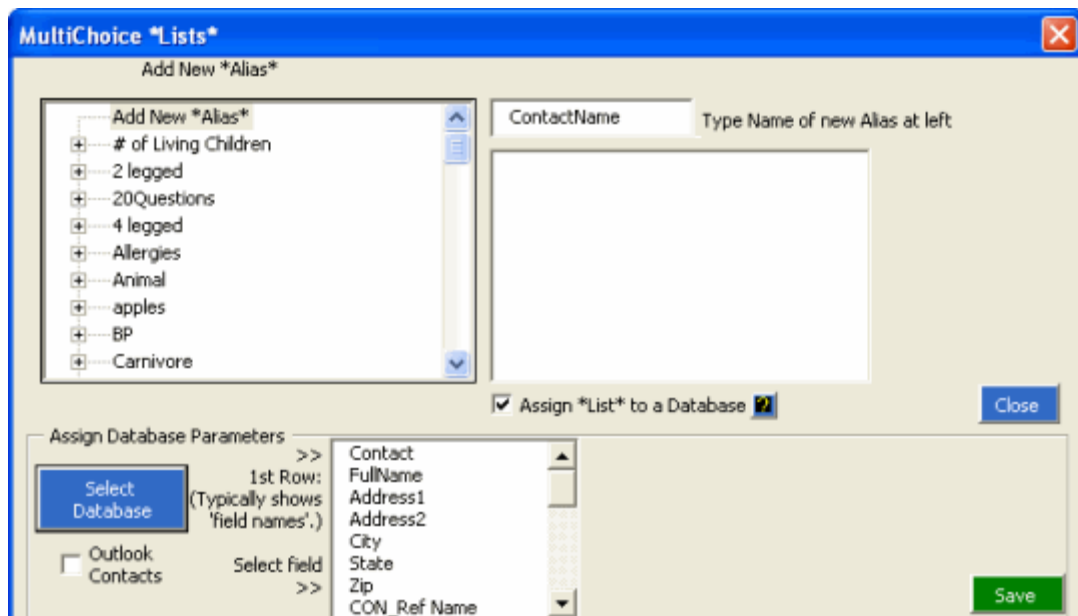


(3) Click the Select Database button. (If you are going to connect to an Outlook Contacts list, check the Outlook box that appears beneath the Select Database button.)

If not Outlook:

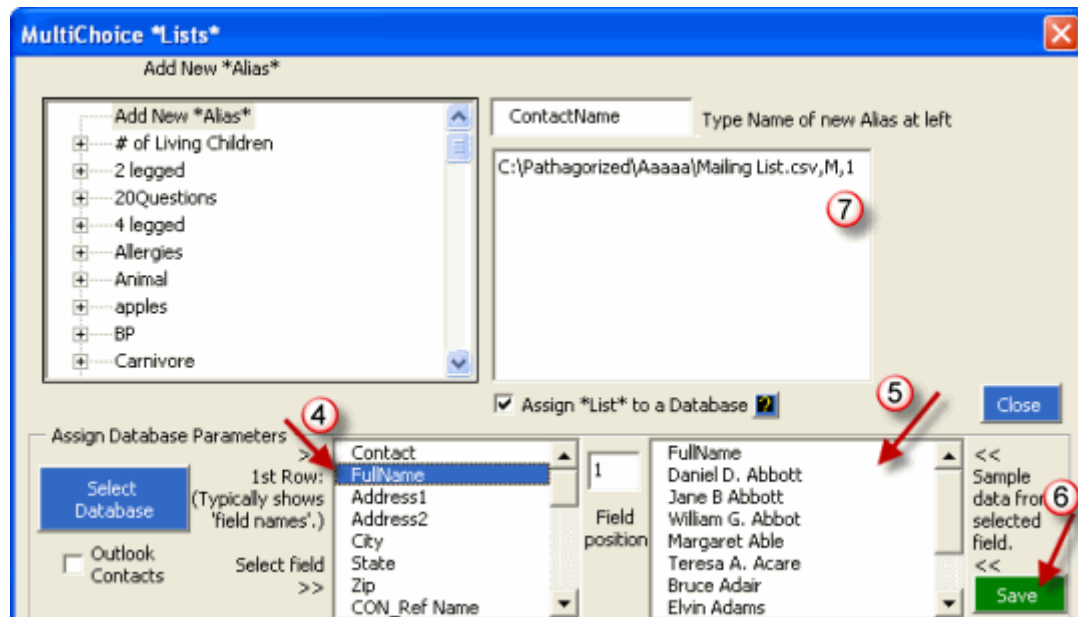
You will be directed to navigate to the Excel, .csv or .txt file to which you wish to connect.

Upon selecting the database, Pathagoras will display in a new list on the screen the contents of the first row of the database. Typically, the first row contains the name of the various fields of the database. Otherwise, it will contain the values of the first record. The screen will look like this:



Select the field that you want the Alias to represent. For this example we select the FullName

entry. The screen changes to this:



(4) Click on the field you want the alias to reflect. A list of the values appears at (5). This display is just to confirm that you have the right list. You do not select anything from it.

(6) If all is 'okay,' click the 'Save' button to lock in the reference. The essential linking information is transferred into the box (7) that otherwise would contain the hand typed list of choices.

You can create more aliases or exit the routine.


If Outlook:

If you wish to connect to Outlook contacts, check the 'Outlook Contacts' box before clicking the Select Database button. You will be redirected to a new screen where you can select the contacts fields you wish to be able to draw from. (You cannot navigate to, or directly link to, the Outlook files.) Once you have made your selections from Outlook, a new file containing only those fields will be created. Once that file has been generated, you will be returned to the original screen and will see the list described in the above paragraph containing the names of the various fields from you Outlook contacts. The remaining steps are similar to those outlined above

Using the alias:

Once you have assigned an alias to a database field, simply type the MultiChoice alias at the appropriate locations within your source document. It is set up just like you set up any other variable or options text block. The only additional requirement is that you must enclose the alias between two '*'s. (It is these asterisks that signal Pathagoras that an alias is being used. Save the source document.

When you recall the source document during a document assembly session, Pathagoras will open the appropriate MultiChoice *List* and present all of the choices the list represents.

 You can link two or more fields in two distinct way:


1. **Using a !group! name.** Here is an example of four fields that together make up a 'full name' (the !CUS! is an arbitrary name for the group. It stands for 'Customer'):

```
[!CUS!*FirstName*] [!CUS!*MI*] [!CUS!*LastName*], [!CUS!*Suffix*]
```

When you press Alt-D, the available values of each variable will be displayed on the Instant Database Screen. When you select one item from any list, the corresponding values from the other lists containing the same !group! value will automatically be selected. [Click here to read more about *Lists* and !groups!.](#)⁴⁷⁶

2. Using a Virtual *List*: You can concatenate up to 6 fields in the MultiChoice *List* screen. The only additional restriction is that the fields must be from the same database source. See [Virtual *List*](#)⁴⁸⁴ for illustrated examples.

24.11 'Virtual' *List*

 In the previous section, we discussed how you can cause variables found in separate Alias *Lists* to act in unison. It is accomplished by pre-pending each alias with the same !group! name.

You can duplicate this 'in unison' behavior in with *List* derived from external data sources in an even more remarkable way. We call this a "Virtual *List*". You simply combine several Alias *Lists* terms that derive from a single data source into a single 'virtual' term.

Let say you have data aliases for your customer list called "FirstName", "MI", "LastName" and "Suffix" (matching fields in your external database).

- **"Standard" approach.** With the above aliases, you can structure your letter in such a way as to bring each variable/alias one at a time. The first line of an 'inside address' to your letters may look like this:

```
[*FirstName*] [*MI*] [*LastName*]
```

- **"Standard" approach, with !Groups!.**

To make sure that everything moves in tandem (so that when you select "[*First Name*]" from the Instant Database screen, the other elements are properly selected, you should assign a !group! name to each element. So, if we pre-pended the group name !client! to the various aliases, the above line of text in your document would actually look like this:

```
[!client!*FirstName*] [!client!*MI*] [!client!*LastName*]
```

The above should still quite understandable to your end-users. However, but the more you add, the more real estate is taken by the various variables, and the more confusing the overall look might be to your end-users.

It is time to consider the 'Virtual Alias' approach.

- **"Virtual Alias" approach.** Pathagoras allows you to create a 'virtual' alias in lieu of the individual 'real' terms. Let's say we want to combine the three components of a full name into the virtual alias 'FullName'. Here is how:

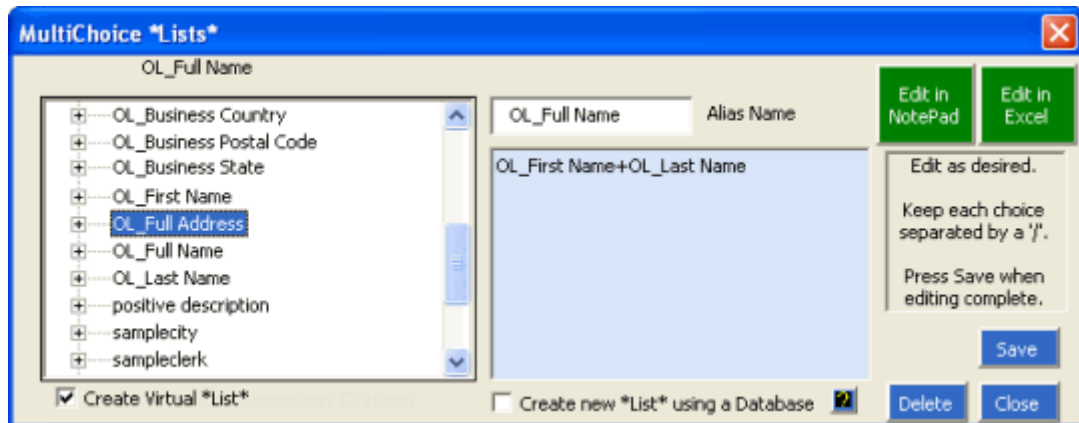
Manual Method:

1. Open the Alias *Lists* editor.
2. Click "Add New Alias"
3. Add a new alias Call the term "FullName"

4. Clear the blue 'formula' box. Then "FirstName+MI+LastName" into the box.
(These are the names of the individual fields that make up the virtual term. NOTE: Spelling is critical! However, Pathagoras will make sure that all components of the virtual *List* exist before creating a virtual term for you.)
5. Save the new alias.

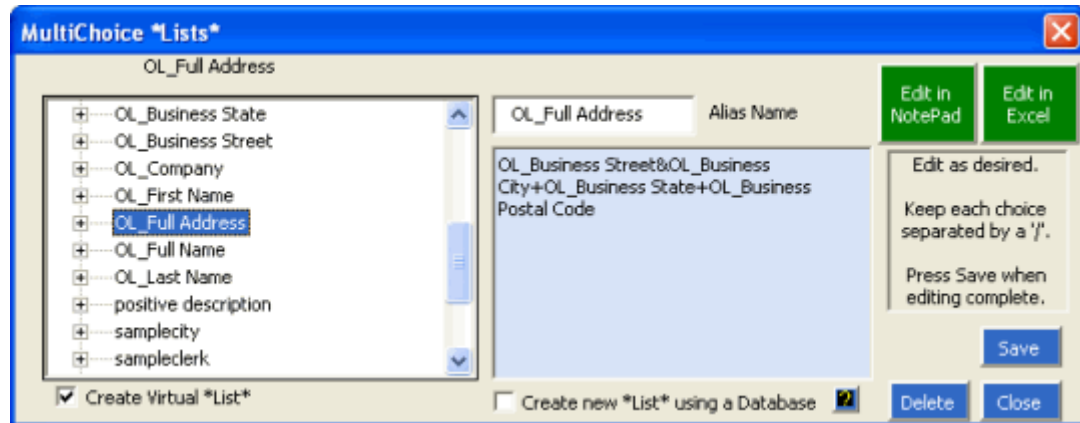
Double-click method:

1. Open the MultiChoice Term editor.
2. Click "Add New Alias"
3. Add a new alias Call the term "FullName"
4. Check the checkbox just beneath the 'tree' (at the left) called "Create Virtual List".
5. Clear the blue 'formula' box (a double click will actually do it). Double-click on the first term that will create the virtual term. The value of the term will be transferred into the formula (blue) box.
6. When you have finished building your Virtual *List*, save it under the new alias.



Use the virtual alias in the same fashion as you would use any other MultiChoice term. That is, you would place the alias within your document where ever you want it to be used. Surround the alias with asterisks to tell Pathagoras that it is a MultiChoice *List* item. Then surround all of that with the proper 'type of term' indicator: square brackets if a variable or curly braces if options text, etc.

Separators: Pathagoras recognizes two separators. The '+' sign (the default) inserts a space between terms. The '&' sign will insert an 'Enter'. So, following the above examples, a full name is likely to be written "FirstName+LastName" (as shown in the above image but an address block will look like this:



i Currently, punctuation is not provided in virtual alias blocks. This is a feature on which we are working. If punctuation is required, you can manually add it after the document has been assembled. Alternatively, simply don't use an alias. Rather insert in the source text the appropriate variables using !group! links and the appropriate punctuation. E.g.,

"[!group1!*City*], [!group!*State*] [!group!*Zip*]"

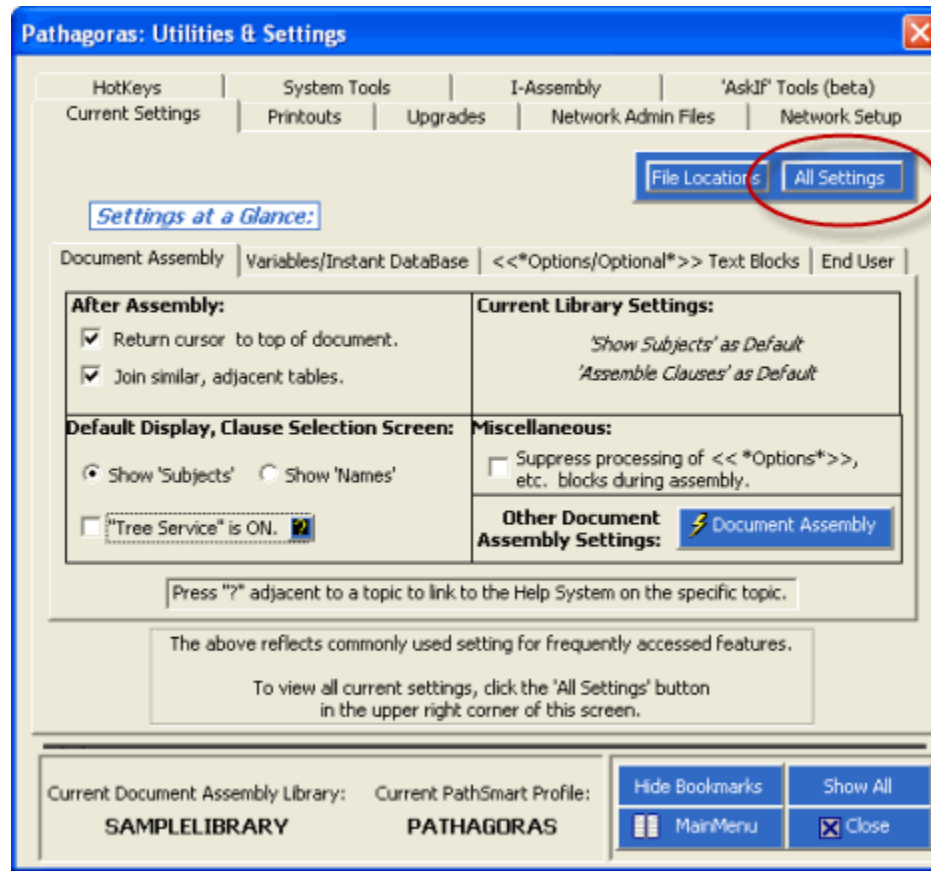
Alias Names: Of course, for Virtual *Lists* to properly identify the base alias names, the base names cannot contain any '+' or '&' signs. (Pathagoras could not distinguish a "&" that was in the base name and one that you added as a separator.)

24.12 Locating, Moving, Pointing (to)

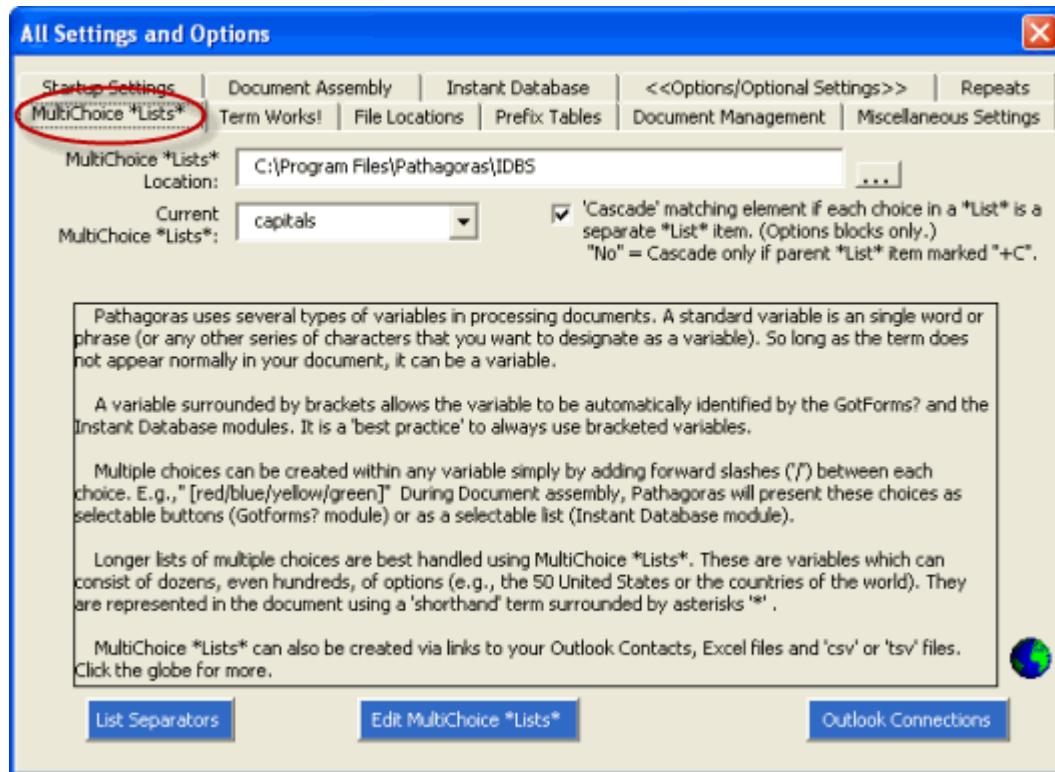
In most cases, the files containing your *Aliases* 'travels' with your Instant Database records. But this is just the default action. You can store your *Aliases* anywhere you want and share them with anyone.

The *Alias* names and the multiple choice lists they represent are stored in a file called "multichoice.csv". Initially, this file is located in the folder c:\program files (x86)\pathagoras\idbs". (The folder "idbs" is where your Instant Database records are stored.)

To move (or find, if missing) your MultiChoice Lists using Pathagoras, click the All Settings button from the Utilities/Settings screen.



From the resulting All Settings screen (shown below), click the *Aliases* tab. This will display the current location of the multichoice.csv file. (At least this is where Pathagoras thinks it is. If the file was manually moved, or otherwise intentionally or accidentally relocated, this is the last location recorded in your Windows registry.)



To move, or repoint to, the *Alias* List, click the "... " button. Navigate to the folder that either contains (if you are repointing) or that you want to contain (if moving) the *Alias* List.

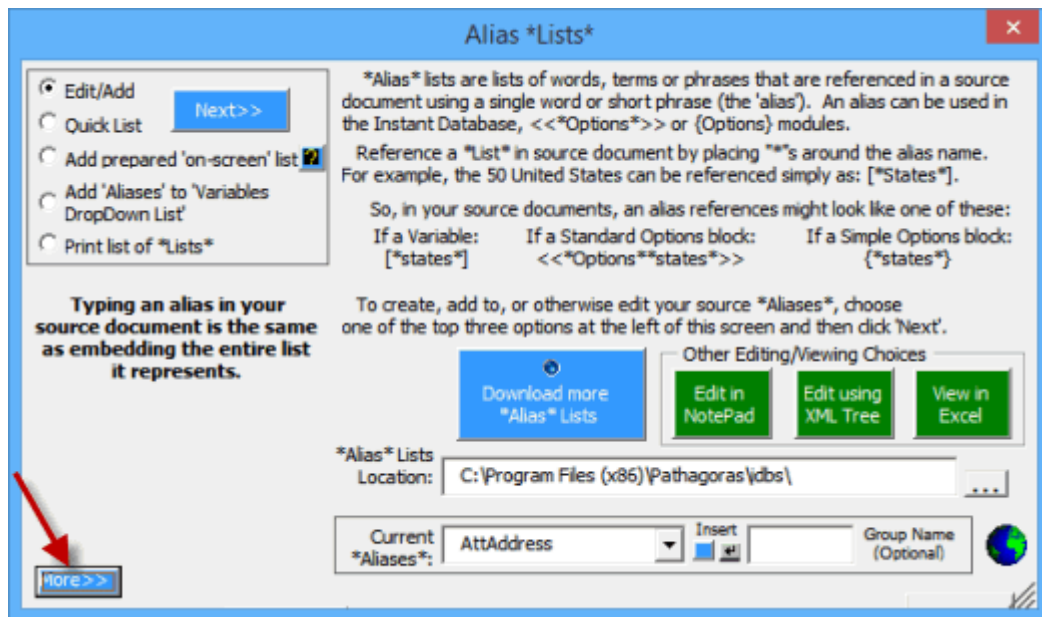
24.13 Alias Table (Embedded)

Typically, your Aliases are stored at you system level (either on your computer or if you have networked everything, on your server) in a file called "multichoice.csv". They can also be saved at Excel files (.xls or .xlsx). [See this link.](#)⁴⁶⁶

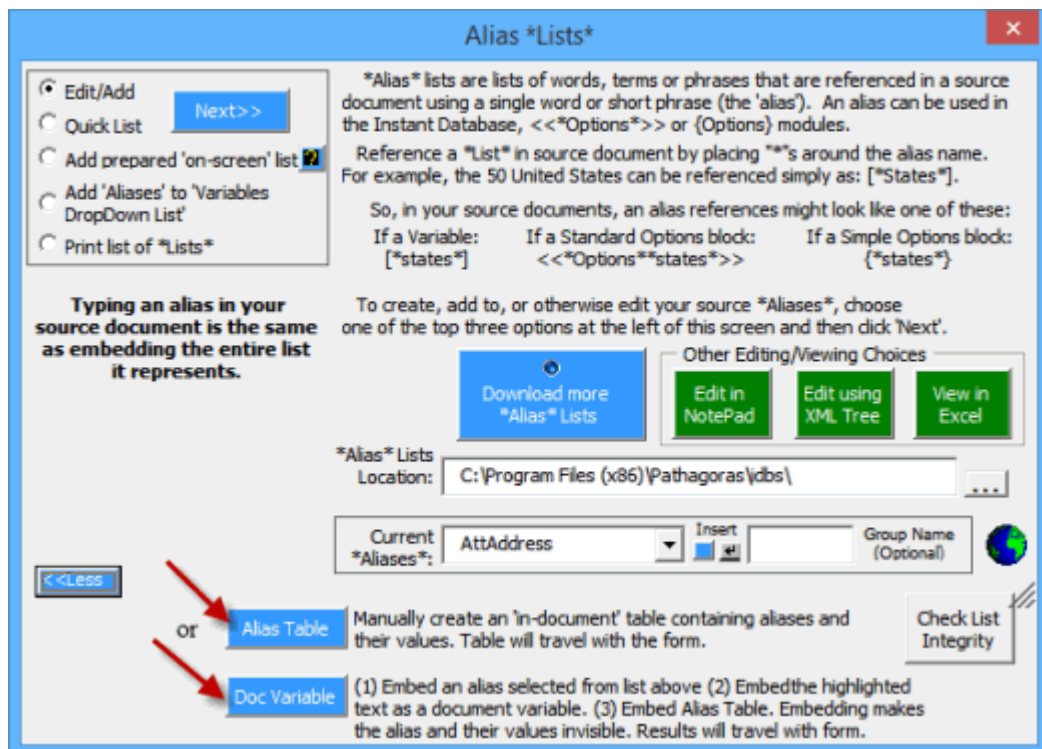
However, if that file is not available to all users, the call to an *Alias* will fail. (This can happen when users are from different offices or are not otherwise pointed to a central server.)

To overcome that 'problem' and make your aliases more 'portable,' you can store an Alias (or several Aliases), along with the multiple choices that the Alias represents, within your document. Pathagoras allows you to paste the aliases you are using into the current document into an '*Alias* Table'. The table sits at the bottom of the document and can be easily seen and edited by the recipient user. (It can also be easily removed before sending to a client or customer.)

To create the Alias Table, display the Alias *Lists* screen (Pathagoras Features | Editing Tools | Alias *Lists*). Click the More button to expand the screen.



Choose the alias you want to embed in the document from the Current *Aliases* drop down list and then click the Alias Table. (You can alternatively embed the alias and their values as hidden Document Variables. This is a good choice only if you know you or the recipient will not need to edit the multiple choice values for the Alias.)



24.14 Sharing Your Lists with the World

Give

You can help to improve the world and make a difference in the lives of others by sharing

Alias *Lists* that you have created. Just send 'universally usable' lists that you have created to:

lists@pathagoras.com

I will post what you send to the website (the page noted below) where it can be downloaded and used by others.

Receive

Conversely, you can download lists that have already been posted to the website. Here are the steps:

1. Visit this page: www.pathagoras.com/multichoice/paths.html.
2. Scan the current listing. If you see one that you like, highlight and copy it;
3. Return to Word and paste the list onto a blank page;
4. Highlight the entire list;
5. Click the Pathagoras features menu and select Authoring/Editing Tools;
6. Select Alias *Lists* from the menu;
7. Click the 'Add prepared 'on-screen' list' from the upper left quadrant;
8. Click Next. Give the list appropriate 'alias'. Click OK. That's it.

24.15 Repairing Alias *Lists*

Occasionally an Alias *List* will go bad. You will know it is bad when you start getting errors (especially 'End of File' errors) But you don't have to shoot it. Just repair it.

Pathagoras has a tool to help determine what in the *List* is bad. Just display the Alias *List* screen, and then click the More button. In the expanded screen, press the "Check List Integrity" button. Pathagoras will report the potential 'bad stuff' it sees.

Unfortunately, due to the nature of some programs (and because a false positive may have been detected), Pathagoras cannot automatically clean the file for you, though. You must open the file, study the reported 'error' and perform any required cleaning yourself. You will need to display the Alias file (called 'multichoice.csv') in Notepad to perform any repairs. This simply requires that you click the green "Edit in Notepad" button. Check for other potential errors while you are there. You should be able to readily discern the 'pattern' for the creation of an alias and its children.

The Pathagoras System

Document Logic

Part

XXV

25 Document Logic

How do some programs seem to 'know' that if Condition A exists, then clauses 1, 2 and 3 should be inserted (or that new sub-question C posed to the end user), but if Condition B exists, only clause 2 and 4 should be inserted (or sub-question B posed to the end user)? Do these programs somehow possess an intuition-like quality?

Well no!

They 'know' what to do because somebody sat down and programmed appropriate logic into the document necessary to make those 'decisions.'

'Document logic' (also called a "Menu Driven Document Assembly") is a wonderful tool, and can make the document assembly process much easier than having to select each clause one-at-a-time. It is document logic that, to many people, make or break a document assembly project. But document logic can be tough. It requires an understanding of a new 'language,' and lots and lots of testing to make sure that the various commands and 'equations' are properly constructed. Nowhere is "GIGO" (garbage in, garbage out) more applicable.

Some document automation programs require document logic as a necessary part of their basic setup. No embedded logic, no action! That drove many people to forgo implementing document assembly from the 'get go.' Not so with Pathagoras. If you want it, its there. Don't want or need it, don't worry about it.

Because Pathagoras is 'plain text,' its form of document logic (and your ability to create and edit the logic equations on the face of the document) is unparalleled Here are some of the benefits:

- You can see the logic equation by just looking at the document. Everything is facial. This is exclusive to Pathagoras.
- You can edit the logic equation on the face of the document itself.
- Since only plain text characters are used, you can edit the logic equation straight from the keyboard.
- Since everything is facial and easy to edit, it is also easy to test, reedit, retest, etc., in many cases without ever closing the document.
- You can test bits and pieces of document logic by simply highlighting, cutting and pasting the section you want to test into a new document. (Other programs require a link to an ancillary document, or embed additional code into the parent document. This prevents this kind of 'small snippet' testing.)
- You can even edit logic equations on a machine on which Pathagoras has not been installed! (Of course, to *test* your work, you will need a Pathagoras computer.)
- Because only plain text characters are used, you can 'paint' the various elements of the equation so they stand out and apart from each other. (Pathagoras has a built in 'painter' that will do this for you.) Hidden fields are always plain grey. (Let's offer a bit of information right here. On the following pages, you will see 'painted' elements of various equations. The colors are for illustration and sample purposes. While they should be helpful to you, the colors are meaningless to Pathagoras -- and are certainly not a required part of logic equation building -- and will not be carried into any final document.)

- Document logic can be implemented from the simplest to the most complex levels. It is not an all or nothing proposition with Pathagoras. Start simple and the build to as fancy and complex as you want.

Let's examine several 'document logic' approaches offered by Pathagoras, from the simplest to the most complex.

(We discuss our philosophy toward Menu Driven Document assembly (aka, Document Logic) in a White Paper so titled. Click here to read.

25.1 Simple Logic

The simplest application of logic within Pathagoras is the selection of one clause over another. While this is not complicate logic, or 'in-document' logic, it does involve the same kind of selection decision that you will be trying to implement as you experiment with the 'in-document' logic features of the program. You have criteria in mind, and make a mental selection based thereon. The same thing happens when you ask Pathagoras to make decisions for you. The only real difference is that you have to express your criteria in writing and in a manner that the program will understand.

The simplest 'programmable' logic blocks are 'Options' and 'Optional' text. They have been fully discussed and illustrated elsewhere (links provided below). We discuss them again here simply to illustrate the heirarchy of document logic expressions available in Pathagoras.

{Optional} and {Options} text using 'simple' enclosures '{' and '}':

{Optional} text: If Condition A (let's say, 'minor children'), the keep 'this' text, otherwise delete it."

{I direct that any funds otherwise due to my minor children be placed in an account established under Uniform Gifts to Minors Act.}

When an {Optional} text block is encountered during the assembly process*, Pathagoras will stop, highlight the text and ask you if you want to keep or delete that text. The positive or negative response to the question controls what happens next: either the highlighted text is retained or it is deleted.

Click here for [information, instructions and illustrated examples](#)¹⁶⁴ on the use of {Simple Optional} text blocks.

{Options} text: Instead of the single action 'keep/delete' logic of Optional text, Pathagoras allows you to set up a collection of choices. When the end user encounters the Options block, the choices are presented separately and the selects one of the available choices. The selected choice remains in the document and the others are excised.

Here is a simple example:

Your package will be shipped by {First Class Mail/**OR**Federal Express/**OR**UPS\\}.

The choices can be small elements as shown above, entire paragraphs and sections, or entire documents. There really is no limit as to the length of the individual choices.

Click here for [information, instructions and more examples](#)¹⁶⁴ on the use of {Simple Options} text blocks.

Robust <<*Optional*>> and <<*Options*>> text.

Robust blocks let you add questions and prompts to help guide the user to the desired choice.

<<*Optional*>> text: If Condition A (let's say, 'minor children'), the keep 'this' text, otherwise delete it."

<<*Optional*Are there Minor Children?*I direct that any funds otherwise due to my minor children be placed in an account established under Uniform Gifts to Minors Act.>>

When an <<*Optional*>> text block is encountered, Pathagoras will stop, highlight the text and ask you if you want to keep or delete that text. If, as above, a question is posed within the administrative section of the text, Pathagoras will ask the question. The positive or negative response to the question controls what happens next: either the substantive text is retained or it is deleted.. Language is retained or language is deleted.

Click here for [information, instructions and illustrated examples](#)¹⁴¹ on the use of <<*Optional*>> text blocks.

<<*Options*>> text: Instead of the single action 'keep/delete' logic of Optional text, Pathagoras allows you to set up a collection of choices. When the end user encounters the Options block, the choices are presented separately and the selects one of the available choices. The selected choice remains in the document and the others are excised.

Here is a simple example:

Your package will be shipped by <<*Options*USPS/FedEx, Next Day/UPS, 72 hours*Your order will be shipped by First Class Mail/Your order will be shipped today for next day delivery by Federal Express/Your order will be shipped today for delivery within 72 hours by UPS.>>

The choices can be small elements as shown above, entire paragraphs and sections, or entire documents. There really is no limit as to the length of the individual choices.

Click here for [information, instructions and more examples](#)¹⁴¹ on the use of <<*Options*>> text block.

'Options' and 'Optional' text blocks as discussed above are placed 'locally'. This simply means that the equation is placed at the precise point where you want the result to be. The logic block is processed as and where encountered.

Other logic blocks are 'remote,' means that the equation is placed elsewhere. A 'remote placement' is typically made in an 'Ask Menu' so that all questions can be presented at one time. But 'remote' is also descriptive of Options/Optional blocks which contain groupnames. As described in the very next section (titled "Intermediate Logic"), Options/Optional blocks with groupnames will generate an identical 'reaction' in a related Options/Optional block further down in the document.

***The document assembly process in Pathagoras is triggered in one of three ways:**

1. Calling in a document via the Clause Selection Screen
2. Calling in a document via a DropDown List

3. Pressing Alt-P against an open document (This is our least 'favorite' method. If you press Alt-P against an original document and then save it, you will have overwritten -- and potentially ruined -- the original.)

25.2 Intermediate Logic

An answer to a question doesn't have to be just for the surrounding text. Document logic allows that answer to be carried forward to multiple locations throughout the document.

So a selection made at the top of the document can be tied to text that resides the middle and the bottom portions of the document.

Pathagoras handles carry-forward and 'memory' via the use of !Groups! and !GroupNames! (actually these two terms are essentially identical and may be used interchangeably).

!Groups! are used in many different areas of the program and have been discussed in several other contexts before now. See, e.g., [!Groups! and the Instant Database](#)¹¹⁹.

A !Group! is any set of <<*Options*/*Optional*>> or {Options/Optional} blocks which carry a common !GroupName!. A !GroupName! is simply a word or short phrase inserted at the begin of the Options/Optional block between two exclamation marks.

Here are two examples, using the same text as the samples provided on the previous page:

```
<<*Optional*!minors!Are there Minor Children?*I have [NumMinCdn] children who are still
under the age of 18 years.>>
```

... elsewhere in the document

```
<<*Optional*!minors!*I direct that any funds otherwise due to my minor children be placed in an account
established under Uniform Gifts to Minors Act.>>
```

If there are other Options or Optional blocks else where in the document with the same !GroupName!, they will be processed identically.

Read more about [!GroupNames! at this link](#)¹⁵¹.

A reminder about the 'standard' document assembly sequence of events is appropriate at this stage. After a document is created via the Clause Selection Screen or inserted onto the screen via a DropDown List, Pathagoras will automatically search for each <<*Options*/Optional*>> block in the document, starting from the top and working down to the bottom. Once the end user makes a choice regarding each block, Pathagoras processes the choice throughout the document. If !Group Names! are used, the decision regarding that particular text block is applied against each member of the !Group! wherever located in the document. Then, Pathagoras returns to the top of the document and searches for another <<*Options*/Optional*>> blocks. It processes it, returns to the top, etc.

When there are no more <<*Options*/Optional*>> blocks to process, Pathagoras 'stops', presenting to you what is hopefully a perfectly tailored initial draft of your document.

Here is another (more self contained) examples (you can copy and paste each example provided into a document. To process the text, press Alt-P.

Thank you for selecting us as your interior design team. We suggest that the color for the living room be <<*Options*!colors!*red/yellow/burnt orange>> and the contrasting trim color be <<*Options*!colors!*sea green/purple/royal blue>>.

So in the above example, if 'yellow' is selected for the answer to the first encountered <<*option*>>, then purple will be automatically selected when Pathagoras processes the second block. (Remember, the selection is positional. Yellow and Purple are in the second positions of their respective <<*Options*>> blocks.

Using "<<*Options*>>" text blocks to pull external text into present document.

The examples above contain the actual 'results' text as an integral part of the <<*Options*>> text block. But you are not limited to just internal text. Pathagoras lets you provide as one or more of your "options" references to external documents or glossary terms. So instead of having to type in each piece of your target text, just type in its reference.

Here is an example where text from a particular document will be returned to the screen. (The assumption here, of course is that 'cntr105', 'cntr155', 'cntr197', etc., actually exist. The example is intended to convey that enclosing a reference to a term or document within "<<" and ">>" will cause Pathagoras to seek it out and insert it when called for.)

```
<<*Options*Are the facts A?/Are the facts B?/Are the facts C?/Are the facts D?/
Are the facts E?* <<cntr105>> /<<cntr155>> /<<cntr197>> /<<cntr200>> /<<cntr205>>>>
```

See [Calls to Other Documents](#)¹⁰⁴ for more examples.

i Instead of presenting 'static' choices as shown in the above examples, you can use [variables] and <<*Options*>> text blocks to cascade a series of progressively narrower choices of clauses for selection by the end user. See [Cascading](#)⁴⁶⁸ for more information and a set of real world examples which you can copy and modify

i The techniques discussed above allow the administrator to create questions and 'responses' which lead to the creation of a particular kind of document. Of course, this is a more complex application. It will take some 'real' time to create. But once done, the 'logic' built into the program frees the end user from having to select actual clauses. Rather, the end user need only answer the questions posed. The administrator has 'pre-ordained' the actual text that is brought into the document based on the answer. The result is that uniformity in document production is assured.

25.3 Cascading Logic

This next example introduces 'multiple comparators.' The ability to use the values of two groups to set the value of a third (or to cascade a question) adds a new dimension to document logic. Here is an example. A 'Will' is the objective. The questions posed are "Sex of Client? (Male or Female)", "Is Client Married? (Yes or No)" and "Does Client have children?" With that information, other decisions can be automatically set or other relevant questions posed.

```
<<*AskOptions(radio)*!T-orSex!Testator = Male/Testator = Female*>>
<<*AskOptions(radio)*!Client Married!Client is Married/Client is Single*>>
<<*AskOptions(radio)*!HasChildren!Client has children/Client has no children*>>
<<*If*!HasChildren!="1",<<*AskRepeat*!NumChildren!Number of
Children*>>,!NumChildren!="0(#)">>
<<*If*!NumChildren!="1",<<*AskOptions(radio)*!ChildGender!Son/Daughter>>,>>
<<*If*!NumChildren!="0",<<*AskOptions(radio)*!AllAdults!All children adults/Minor child(ren)
*>>,!AllAdults!="1(2)">>
<<*If*!NumChildren!="0" and !AllAdults!="2",!Guardian!="True",!Guardian!="False">>
```


Comparisons may only be 2 deep (currently).

'Legal' connectors:

'AND' or 'and'

'OR' or 'or'

'+'

'_'

Legal comparators

'='

'~' (not equal)

i We encourage anyone with more complex needs to contact us. Using a 'real world need as a guide will help us to refine this function.

Other examples approaches:

```
<<*<AskOptions<radio>*<Size!Small/Medium/Large/Custom*>>>
<<*<AskOptions<radio>*<Color!Red/Blue/Green/Custom*>>>
<<*<If*<Size!="Custom">AND <Color = "Custom",<<*<AskOptions*<ExtraCharge!$50/$60/$70*>>,<>>
<<*<If*<Size!="Custom">AND <Color ~ "Custom",<<*<AskOptions*<ExtraCharge!$100/$150/$200*>>,<>>
<<*<If*<Size!="Custom OR <Color! = "Custom",<<*<AskOptions<radio>*<Shipping!$20/$30/$40*>>,<>>
```

The symbols in parenthesis are optional, but should be included. In line 4, the (#) indicates that a pure number is being set. In line 6, it indicates the number of choices that the group typically provides. So if 'NumChd' is more than zero, the question AllAdults is posed, otherwise, !AllAdults! is set to 1 (of 2 choices).

Beginning with line 4, values are set or more questions asked. In line 4, the initial If equation that tests !Children! compares it to "=1". This comparison is to the position of the two choices regarding !Children! (set in the preceding line), not the number 1.

25.4 Advanced Logic

Actually, we don't really have any advanced logic built into the program, but we do have workarounds for some classic logic problems:

IFF

If and only if (shortened **iff**) is a 'biconditional logical connective'. Not only must a value or a particular group be true, but all other members of the same group must be false.

You can create the functional equivalent by setting up each of the two conditions in separate If statements. You will need to assign a new value in the Ask table. We'll call it !OnlyChoiceX!

```

1 <<*AskOptions*!GroupXYZ!Choice 1/Choice 2/Choice 3*>>
2 <<*If*!Choice 1!="True" and !Choice 2!="False" and !Choice
3 3!="False", !OnlyChoice1!="True",!OnlyChoice1!="False">>
4 <<*If*!Choice 1!="False" and !Choice 2!="True" and !Choice
  3!="False", !OnlyChoice2!="True",!OnlyChoice2!="False">>
  <<*If*!Choice 1!="False" and !Choice 2!="False" and !Choice
  3!="True", !OnlyChoice3!="True",!OnlyChoice3!="False">>

```

(remember, Pathagoras will assign 'hard values' to selections made via Options. So when the , so the individual choices made when line #1 is processed are converted to groups, and assigned True or False values, depending upon if it was selected. The individual values can in turn be used in Optional equations in the document body.

(letter body . . .)

```
<<*Optional*!OnlyChoice2!This text displays when only Choice 2 is true, and the others are false>>
```

This works best when there are only 3 or 4 values that must be analyzed. It beyond that, the If statements (2 through 4 above) can be long and many.

25.5 Interviews

A typical 'well developed' document will likely have many Options and Optional blocks throughout the document. Some blocks are tied together with !GroupNames!, many are stand alone.

In a typical document assembly session, Pathagoras will identify the various Options/Optional/Repeat blocks starting at the top and working its way to the bottom. Each block is processed as it is encountered. If a !Group! is encountered, all members of the group further down in the document are processed when the first member of the !Group! is answered.

There is only one drawback to this approach: each Option/Optional/Repeat block is answered one at a time, and as where encountered in the document. 'One-at-a-time' is not a 'bad' thing, but after a while it becomes tedious. You may find yourself asking 'Why can't I just answer all the questions at the outset and then let the program take over and complete the document for me?'

You can, and that's where Interviews come into play. Once you have implemented !Groups!, you are ready for the next (entirely optional) step. And that step is to gather all the <<Options/Optional>> blocks in a single location at the top of the document. That way, the questions can be asked as a group (instead of them being asked and processed individually as described on the previous page).

Beginning with the next main section of this Manual, we will introduce the 'Ask' prompts and the parent 'AskTable'. Once you understand the structure of these elements, we'll show you how they can be automatically created and manually modified to meet your needs.

Please note the following. The previous, and subsequent pages discuss document logic from its simplest to the most advanced possible implementations. One is not better than another. If you are satisfied with the simplest iterations, stop there. If you want to move to !Groups! and AskTables, do so at your own pace. Just keep these things firmly in mind.

- Implementing document logic is entirely optional. Of course, the more you implement, the more automated you can make your document for yourself and for others. But nothing is required. Take 'Pathagorizing' and automating your documents and clauses it at your own pace, and don't feel that you have to implement anything else.
- The Ask tables and the Interview Wizard draw their information from existing <<*Options and Optional*>> text blocks. Not all questions need be in the Ask Table for the system to work well.
- Everything builds on top of everything else. An Optional text block is augmented by the insertion of a "!GroupName!", not replaced by it.
- All features described in the above sections work whether you are adding simple logic to an entire document (typically called a 'template') or to individual clauses that you intend to cobble together from scratch using the Document Assembly/ Clause Selection Screen routines.

25.5.1 The <<*Ask. . .*>> Commands

Advanced: <<*Ask. . .*>> Commands

Sometimes you want to <<*Ask*>> a question of the end user: Something like "Are there children?" or otherwise "Has a certain condition been met?" Based on the answer provided, you want the program to (1) select text to be inserted or (2) set, or ask to be set, another value.

The <<*Ask. . .*>> commands ask those questions.

<<*Ask. . .*>> commands are siblings to <<*Options*>> and <<*Optional*>> text blocks within a document. The difference between the two is that the <<*Options*>> and <<*Optional*>> text blocks contain substantive text. <<*Ask*>> blocks do not. An <<*Ask. . .*>> command exists simply to ask a question and record the answer for later use. Once answered and recorded, the <<*Ask. . .*>> command is deleted.

The 'secret' to how and why they work is that the answer to each <<*Ask. . .*>> is assigned to a !groupname!. It is the !groupname! that ties values together.

Types of <<*Ask. . .*>> Commands:

Four <<*Ask. . .*>> **Commands** are available in Pathagoras. Collectively they cover the range of possible 'Asks' that an author would need to implement a full Interview (Read more about the structure of Ask commands in the below article called <<*Ask*>> Elements):

<<*AskOptions*!Children!No children/One child/2+children*>>

<<*AskOptional*!Minors!Are any children <18 years of age?*>>

```
<<*AskRepeat*!NumChildren!How many children?*>>
```

```
<<*AskValue*!Score!What score did student receive on test?*>>
```

There are three other important commands that bring <<*Ask. . .*>> commands to their full potential.

1. The <<*Set*>> command. The Set command can be used to hard code a value in the Ask table area of the document..

```
<<*Set*!Children!=0(#)*>>
```

2. The <<*If*>> command. This is a powerful tool that can be used to set a value to a !groupname! based on an answer given in response to an <<*Ask. . .*>> command, or to pose another <<*Ask. . .*>>. Here is an example:

```
<<*If*!Children!="Yes", <<*AskOptional*!Minors!Are there minor  
children*>>, !minors!="False">>
```

3. The <<*Case*>> command. Another power tool used to analyze complex Ask structures involving comparison of multiple elements.

The [Set](#)⁵⁰⁸, [If](#)⁵⁰⁶ and [Case](#)⁵¹⁸ commands are discussed in separate sections below in this manual.

Here are a few additional things you should know about <<*Ask. . .*>> commands

- They are totally optional. If the first appearance of an <<*Options/Optional*>> block in the document provides a prompt that is clear enough for the typical user to make the proper selection, you may choose to not use an <<*Ask*>> command.
- If used, <<*Ask. . .*>> commands are typically placed at the top of the document. That way, all Asks can be posed at the start of the assembly process. (It is also easier to edit questions you will be asking your end users when they are grouped together.)
- <<*Ask. . .*>> commands are all 'plain text'. As such, you can easily copy a good set of <<*Ask*>> commands from one document to another. Doing so will help you to make the questions uniform from document to document.
- If you use <<*Ask. . .*>> commands, you can get rid of the prompts that you may have typed within your <<*Options/Optional*>> blocks. This may result in a significant savings of document 'real estate'.

Placement of <<*Ask* . . .>>, <<*If* . . .>> and <<*Set* . . .>> commands.

<<*Ask* . . .>>, <<*If* . . .>> and <<*Set* . . .>> commands are typically located at the top of the document. That way, it is the first thing that the end-user will encounter during a document assembly session.

However, these commands are typically created *last* in the **source document** creation process. The reason is this: unless you know what the <<*Options . . . Optional . . . Repeats*>> text is, you will not know what to even "Ask".

So as you study and try to mimic the examples on the following pages, keep the above in mind. We recommend that you don't even try to create the <<*Ask*>> commands until you have thoroughly composed and tested your document. Remember that <<*Ask. . .*>>, <<*If* . . .>> and <<*Set* . . .>> commands are

optional, so you certainly don't need them in the initial stages of source document development.

NOTE: Regardless of placement, each `<<*Ask* . . >>` and `<<*Set* . . >>` command must be on a separate line.

Benefits of `<<*Ask*>>` commands:

You can create better questions for the end users. (A typical Options block may or may not contain an answer that is sufficiently descriptive. Sometimes the Options block is simply `"<<*Options(radio)*!Client!*him/her/them>>"`. The choice presented to the end user is, therefore, "him" "her" and "them" with a reference to "Client" in the Group box at the top of the selection screen. The `<<*Ask*>>` command can read something like `"<<*AskOptions(radio)*!Client!Our client is a male/Our client is a female/Our client is more than one person*>>"`) (See ['Hover Over' text](#) ⁵⁰¹ below for even more functionality.)

The experienced user will observe that any `<<*Options*>>` block (including the one above) can be set up to provide the identical prompts as may be provided in the `<<*Ask*>>` command. While true, we still recommend the use of the `<<*Ask*>>` command. Here are several reasons why:

When you add prompts to an `<<*Options*>>` block, the amount of 'real estate' it consumes can be substantial. When that block appears in the middle of a paragraph, it can be difficult to visually process (both to the editor and the end user who may peruse the document before processing it.)

There is something intangibly better about compositional questions being at the top of, and segregated from the body of, the main text.

Best reason: Once you have your 'best' questions composed and residing nicely within one of your documents, you can copy and paste an entire collection of `<<*Ask*>>` commands into any other document containing similar `<<*Options/Optional*>>` blocks. This assures a consistency in the questions that are posed to the end user.

For documents-by-building-block aficionados, you can also save the AskTable as a separate clause, calling it in along with other clauses to build, and then process, the document.

Try that with other programs!



Hover-Over text: In addition to 'prompt' text that will appear on each 'choice button' on the screen which presents the various options, you can provide up to 256 more characters of 'hover-over' text for each choice presented. This hover-over text will appear when the user moves the cursor over each choice. It will display only during the time the cursor is 'hovering over' the selection. To add hover-over text, simply type a '+' sign at the end of the prompt text and then type the hover-over text. Make sure the hover-over text is within the administrative section of Options block (i.e., before the third asterisk).

25.5.2 `<<*Ask*>>` Elements

An `<<*Ask* . . *>>` command requires the following elements:

- Opening `'<<'` and closing `'>>'` to designate the scope of the command. (Remember: the red and blue colors used in these examples is not important. We add color only for emphasis and clarity.)

- The term "***AskOptions***", "***AskOptional***", "***AskRepeat***", "***AskValue***" (depending upon the nature of the command) followed by the !group! name. Of course, this group name will relate the answer to the ask prompt to Options/Optional blocks in the body of the document.

- **<<*AskOptions*>>** blocks: list the choices you want to present to the end user. Separate each choice with a '/'. Close the list with a '*'.

E.g.,

<<*AskOptions*!shipmethod!FedEx/UPS/USPS*>>

and later in the document:

Your order will be sent by **<<*Options*!shipmethod!Federal Express (2 day delivery/United Parcel Service (ground; allow 4 days for delivery/US Postal Service (1st class mail).>>**

- **<<*AskOptional*>>** blocks: pose the question that you want the end user to answer. Close the question with a '*'.

E.g.,

<<*AskOptional*!freeshipping!Will shipping be free*>>


and later in the document

<<*Optional*!freeshipping!*Plus, if you place your order in the next 30 days, your shipping is free!>>

- For **<<*AskRepeat*>>** blocks, pose the question that you want the end user to answer. Close the question with a '"'.

E.g.,

<<*AskRepeat*!NumCdn!*How many children?>>

 See discussion on how Pathagoras handles the repeats 'count' with Options blocks beginning [this page](#)³⁹² of the Manual.

- For **<<*AskValue*>>** blocks, you can ask a question that you want the end user to answer. Close the question with a '"'.

E.g.,

<<*AskValue*!Score!*>> or

<<*AskValue*!Score!Test Score?*>>

and later in the document

<<*Options*!Score!0-59/60-69/70-79/80-89/90-100*F/D/C/B/A>>

i Note the placement of the administrative "*" (asterisks). Just like in a regular `<<*Options/Optional*>>` block, the stars close the administrative section. The administrative section includes any questions that are posed. (As used in Pathagoras, the 'administrative section' of a command is set apart by "*" from the text that is intended to remain in the document. In an `<<*Ask*>>` block, it is all 'administrative text' since none of it remains in the document.)

i Each Ask command must reside on its own line. It cannot reside on the same line as another Ask command, nor can any other text be combined with it.

i The Ask commands, collectively, constitute the Interview phase of document assembly. Ask such, the commands should appear at the top of the document, or in a separate document that was written solely for purposes of conducting an interview and calls on other documents or building blocks in response to the interview answers. You should avoid interspersing Ask commands within the document body. Pathagoras will likely not crash if you do, but it is not considered a best practice.

25.5.3 `<<*AskOptions*>>` options

There are several AskOptions 'options' that provide you flexibility in how you present the question to the end user.

The standard `<<*AskOption* . . .>>` will present to the end user a series of check boxes from which the user can select one or more of the options.

`<<*AskOptions(radio)*. . .>>` presents a series of radio buttons from which the user can select only one of the options.

`<<AskOptions(connector)*. . .>>` presents a series of checkboxes from which the user can select one or more of the options, and adds to the bottom of the selection list an optional group of connectors. The selected items are presented in a linear fashion with the selected connectors interposed between the selections. (Note: the connector will be applied only to linear style options.

25.5.4 `<<*AskValue*>>` Command

The `<<*AskOptions*>>` command returns a 'fixed' result based on the position of the 'choice' in the list you provided. If you chose 'bananas' from the choices apples/banana/cherries, Pathagoras only knows that you chose the second item.

Sometimes this 'positional' approach will not work. Sometimes the actual value (and not its position in the list) is needed to control the result. If the answer to a question falls within a range of values, as opposed to being a specific value, you need a different approach. That's where the `<<*AskValue*>>` command comes into play.

Example:

Grades: An 'A' is a range from 90 to 100, a 'B' ranges from 80-89, etc. A score that deserves Honor Roll recognition is any score of 85 and above.

Let's assume a test score of 88. You conceivably could have several lists of 101 choices

each (all tied together with a groupname so they change in tandem), one list representing the actual numeric score (e.g., 100/99/98/. . . /90/89/88/. . . /0), another 100 element list representing the letter grades by position (A/A/A/. . . /A/B/B . . . /F), and 100 choices representing whether a particular grade made the honor rolls (Honor roll/Honor roll/. . . /Not honor roll/. . . etc). But that borders on the absurd.

Enter the <<*AskValue*>> command, Pathagoras 'Asks' for a specific value of 'something' (grade, quantity, score, etc.) at the top of the document. It will then analyze the response per a variety of 'ranges' that you provides elsewhere in the document. As with other <<*Ask*>> commands, Pathagoras requires you to provide a !groupname! to associate it with other elements in the document.

The AskValue method. Example #1

```
<<*AskValue*!Grade!Grade Awarded*>>
```

- a. Your end of semester grades have been released.
- b. Your raw score for the semester was <<!Grade!>>. This translates to a letter grade of <<*Options*!Grade!0-59/60-69/70-79/80-89/90-100*F/D/C/B/A>>.
- c. <<*Options*!Grade!0-84/85-100*/Congratulations! You have made the honor roll.>>

(You can copy and paste the above sample into your document and 'process' it.)

When the user assigns a value in response to the <<*AskValue*>> command, Pathagoras remembers its value and applies what it knows in one of two situations:

1. If an options block containing only the !groupname! (written like this: <<!groupname!>>) is located within the document, it will be directly replaced with the raw value. This is shown in line b. above.
2. When you have provided a 'full' <<*Options* . . . >> block that contains value 'ranges' within the administrative section of the block, Pathagoras will analyze the answer using those ranges and return the appropriate value. This is shown in lines b. and c. above.

Note (and this illustrates the very important distinction between AskValue and AskOptions): The element count need not be the same within various member of the AskValue group member. -- note that 5 options are provided in line b. above. Only 2 choices are provided in line c. (Note also, in line c, the result of the first choice -- the grade is between 0 and 84 -- is a 'blank'. There is no text between the asterisk that closes the administrative text and the first slash, meaning 'type nothing'. Therefore, no 'honor roll' award is presented.)

The AskValue method. Example #2

```
<<*AskValue*!Total!Total of Order*>>
```

Your order totals \$<<!Total!>>. <<*Options*!Total!0-75/>75*Your purchase does not qualify you for free shipping./Congratulations! You are eligible for free shipping of your order. Select your preferred shipping method below.>>


```
<<*Options*!Total!<=74.99/>=75*/<<*Options*!Free Shipping Method!*FedEx/UPS/USPS>>, >>
```


(You can copy and paste the above sample into your document and 'process' it.)

The above example provides a different situation where AskValue might be used. It also illustrates that non-finite ranges can be used. Note the '>75' in both lines. You can use '>' and '<' and '<=' and '>=' (The order of the signs does not matter. So you could have '=>' as well.)

In the area of direct value replacement, you have to decide whether the value is best provided using the Instant Database or via the AskValue command.

When AskValue is better than Variables:

- With AskValue, the presumption is that the value will be analyzed in multiple locations in the document with varying sets of ranges. (This is not a requirement, but is the reason why the routine was written. You can use AskValue to replace AskOptions, but the setup for AskOptions is quite a bit easier.)
- Use AskValue where there is no pressing need to save the value as part of an Instant Database record. You want to save the customer's name, address, etc., so that you can use that information in letter after letter, but you may not need to record (at least for document assembly purposes) the specific grade or the amount of the purchase that leads to the generation of the document. (That data presumably is recorded elsewhere.)

 Note the following structural requirements:

The <<*AskValue*>> command, like all <<*Ask . . .>> commands, closes with a "*".

The comparison must be between a minimum of two sets of values. Those values can be a finite range (e.g., 0-75) or a non-finite range (e.g., >75; >=75 will also work).

Each 'range' option must have a corresponding 'return' option. So, in this example:

```
<<*Options*!Grade!0-89/90-100*Sorry, you did not make the honor roll/Congratulations! You have made the honor roll.>>
```

there are two ranges: '0-89' and '90-100'. The 'return' text is whatever appears after the "*" that closes the administrative section of the block.

Let's provide one more example. There is no sense in 'rubbing in' the fact that the 75 grade student did not make the honor roll. Perhaps it is best just to say nothing.


```
<<*Options*!Grade!0-89/90-100*/Congratulations! You have made the honor roll.>>
```

In the above example, "/Congratulations! You have made the honor roll" is now the return text. The slash, with no text preceding it, means it is a 'blank' choice. It returns 'nothing'.

When a [Variable] is better:

- When the result is a defined value needed only for a direct replacement and need not be

analyzed for use elsewhere in the document and it is imperative that you save the value in the client/customer's Instant Database information.

 **NOTE:** <<*AskValue*>> commands must be manually created in the AskTable. If you have structured an <<*Options*>> block in the body of the document that is being scanned which contains ranges of values, Pathagoras will create an <<*AskOptions*>> command at the top, with the ranges reflected as the alternative choices. Simply change <<*AskOptions*>> to <<*AskValue*>> and either replace the alternative choices with a simple question or just close the block with a "*".

25.5.5 <<*If*>> Command

While <<*Ask*>> commands will constitute the bulk of the document logic table, it is the <<*If*>> command that forms the backbone of document logic process. You will need the <<*If*>> command to declare your comparisons and to set up your decision and branching points.

Let's say you are drafting a document that has optional text dealing with children. You want the text to be kept in the document when there are children and discarded when there are none. Let's further say that the document contains other language that you want to keep when there are *minor* children, but which should be discarded where there are none.

Using AskOptional commands, you might ask the question "Are there children?". If (but only if) that question is answered "Yes", you might then ask "Are there minor children?". (Of course, if there were no children, there couldn't be any minor children; therefore the question should not be asked..)

Here is the way this sequence might appear:

```
<<*AskOptional*!Children!Are there children?*>>
<<*If*!Children!="Yes",<<*AskOptional*!Minors!Are there minor
children*>>,!minors!="False">>
```



NOTE: The values "Yes", "True" and "1" are functional equivalents. The above could have been written:

```
<<*AskOptional*!Children!Are there children?*>>
<<*If*!Children!="True",<<*AskOptional*!Minors!Are there minor
children*>>,!minors!="False">>
```

Let's study the structure.

1. Note first that a !groupname! (in this case !children!) is used. As with other !group!, the name ties the various elements of the document together.

2. Now note the classic three-part "if . . . then" programming structure in line 2:

"If Condition, (If 'True' Action), (If 'False' Action)"

This three-part logic structure is practically universal. Simple commas are used to separate the three elements.

As applied to the example, the user first encounters the initial (Children) question. The user provides either a "True" or "False" (or "Yes" or "No") answer. The "If" command is encountered immediately after. Pathagoras 'measures' the value of the !Children! group, and responds accordingly. If !Children! was answered "Yes" (by checking the checkbox), Pathagoras presents the prompt that follows the first comma. If Pathagoras sees that !Children! was not answered

"Yes", it processes the information that follows the second comma.

It is perfectly acceptable to leave the 'False' portion of the 'If' statement blank:

```
<<*If*!Children!="Yes",<<*AskOptional*!Minors!Are there minor children*>>,>>
```

However, the above statement is incomplete if your intent is to set !Minors! to "No" if the value of !Children! is 'No'. The 'opposite' of the 'True' value cannot be assumed. You must affirmatively set !Minors! to zero (or any other value that you want the False selection to set. The below 'If' statement is correct:

```
<<*If*!Children!="Yes",<<*AskOptional*!Minors!Are there minor children*>>,!minors!="False">>
```

See [Bigger Example](#)⁵²⁷ for more examples of the <<*If*>> command.

A bit of Math:

Pathagoras can do a bit of math for you if you desire. When a !GroupValue! has been set to a numerical amount, either through an <<*AskRepeat*>> or a <<*Set*>> function, you can perform a math function on the values. If the value matches the comparison set up, then the 'True' part of the <<*If*>> is processed, otherwise, the 'False' part is.

```
<<*If*!NumCdn!>5,!CrazyHouse!="True",!CrazyHouse!="False")>>
```

Multiple Comparators

Math (above) lets you add values assigned to !groups!. But what if you just want to ask "If This AND This are true (or If This OR This are true), then do this.

That is possible. Just use the 'AND' or the 'OR' connector:

```
<<*If*!NumCdn!>2 AND !NumPets!>2,!CrazyHouse!="True",!CrazyHouse!="False")>>
<<*If*!NumCdn!>4 OR !NumPets!>4,!CrazyHouse!="True",!CrazyHouse!="False")>>
```

Debugging 'If's

When your 'If' statements seem not to give the expected results, check these elements:

The sequence of the 'True' and 'False' portions

The sequence of a precedent Ask commands that sets the value of the 'If' comparator.

The proper spelling of the !GroupNames!.

Is the 'False' value of the statement set? [See above](#)⁵⁰⁷ for discussion.

NOTES:

1. The 'If' command is one of the 'Ask' commands (just without the 'Ask' prefacing it). We could have used 'AskIf' as the key word, but decided to use just the two letter 'If.'
2. The 'If' command, and it's various parts (the 'True' part and the 'False' part and its mandatory structural elements) is definitely 'programming language. Therefore, If's will be use only within the Ask table portion of a document where programming is allowed.
3. Pathagoras does not use programming language (e.g., 'If') in the body of any of 'Pathagorized' document. If the block of text within your document is 'optional', it must be part of an <<*Optional* . . . >> text block. Do not try to program within the document by creating an <<*AskOptional* . . . >> command at the top of the document, and then measuring the answer with an <<*If* . . . >> block in the body.

4. Use only <<*Options* and <<*Optional* . . .>> blocks in the document body. They can all stand alone. If you need to tie the various blocks together (so that a selection of one results in the selection of a related other, use a !groupname!. It is groupnames that are a functional equivalent of 'If' without the necessity of programming a Boolean formula. (It is the groupname in the <<*If* . . . >> command that allows it to control the result of <<*Options* . . .>> and <<*Optional* . . . >> blocks further down in the document.

25.5.6 <<*Set*>> Command

Sometimes you need to **set** a value within an Ask table. To do so, use the <<*Set*>> command.

When <<*Set*>>-ing values, you are more or less manually setting values associated with a !groupname!. This is precisely what Pathagoras does when it processes a selection you make from an AskOptions or AskOptional command.

The formula is "<<*Set*!groupname!=" followed by a value. The value can either be "True", "False", "Yes", "No", "1", "0" or any equation using previously set groupnames. See dog and cat example below.

Consider this:

```
<<*AskOptional*!Married!Is our client married?*>>.
```

When answered, the value assigned to !Married! is either "True" or "False" (or "Yes" or "No" (or even more accurately, "1" or "0").

You can directly set this value (if the situation so calls) in the following manner:

```
<<*Set*!Married!="True">>
```

```
<<*Set*!Married!="1">> is the functional equivalent of the above line.
```

A more likely application for the use of <<*Set*>> commands is when you want to compare or add (or subtract) the values of one or more existing !groupname! values.

E.g.,

```
<<*AskRepeat*!NumCats!*How many cats?>>
```

```
<<*AskRepeat*!NumDogs!*How many dogs?>>
```

```
<<*Set*!NumCritters!=!NumDogs!+!NumCats!>>
```

You can also use a value set by an <<*AskOptions*>> or <<*AskOptions(radio)*>> command to set another value. Here's the setup, based on two MultiChoice lists. The alias of the first MultiChoice list is *States*. It contains 50 elements (the 50 United States of America). The second alias is called *SecType*. It, too, contains 50 elements. The values of these elements are the type of security instrument ('mortgage' or 'deed of trust') associated with each of the 50 States.

The AskOptions below (first line) sets the value of the groupname !st! to the value the user selects from the list of the 50 States when presented in the Interview. Once that value is selected, Pathagoras sets the value of the groupname called !Security! to the parallel position as the selected state. (So, if Indiana, the 14th position in the list of *States*, is selected, the value of !st! would be '14'. Then Pathagoras processes the !security! groupname by looking at the !st! position in the SecType list. "Mortgage" occupies that 14th position, and therefore Mortgage is assigned to !security!

```
<<*AskOptions(radio)*!st!*states**>>
```

```
<<*Set*!security! = !st!*SecType**>>
```

Less uses of the <<*Set*>> command are the following examples are for advanced users. These users can hard code the values of more complex AskOptions choices so as to pass selected values into the Instant Database tables:

Options (single choice):


<<*Set*!Spouse!="2(2)">> (The first number reflects the selected position of the choice, and the number in parentheses represents the total number of choices)

Options (multiple choices):

<<*Set*!Spouse!="1,3,5(9)">> (where the first numbers, separated by commas, reflect the selected positions of the choices, and the number in parentheses represents the total number of choices)

Numeric/Repeats:

<<*Set*!NumCdn!="5(#)">> (where the first number reflects the number or repeat value, and the pound sign/hash mark tells Pathagoras that an actual number, as opposed to position, is intended.

 Note: the <<*Set*>> command is a 'stand alone' command. It is not used in conjunction with an <<*If*>> (shown in previous screens). <<*If*>> equations allow groupnames to be set directly. E.g, see this example from the previous page:

<<*If*!NumCdn!+!NumPets!>5,!CrazyHouse!="True",!CrazyHouse!="False")>>


25.5.7 Creation of Interviews

Enter topic text here.

25.5.7.1 Automatic Creation of <<*Ask*>> prompts

Automatic creation of <<*Ask*>> commands:

At your request, Pathagoras can scan your source document for any <<*Options*>>, <<*Optional*>> and <<*Repeat*>> blocks that are present. It can pull the essential elements from each such block and then create a basic <<*Ask*>> command for each such block, placing it the top of your document. You can further edit and refine the prompt so that it best meets your needs.

 Note: Only <<*Options*>>, <<*Optional*>> and <<*Repeat*>> blocks that contain a !group! reference will be processed. So, before running the routine discussed below, you may wish to add a !group! reference to your <<*Options*>> and <<*Optional*>> text blocks to take advantage of this feature. Even if there is only a single member of the group, you may find that the benefits of the <<*Ask*>> command are worth the effort to do so.


To create the <<*Ask*>> commands in this automated fashion:

1. Make sure that the document to which you want to add <<*Ask*>> commands is on screen.
2. Click the Pathagoras | Authoring/Editing Tools | Create Ask Blocks element.

When clicked, Pathagoras will scan the document for any <<*Options*>> and

<<*Optional*>> text blocks that contain a !GroupName! reference. Pathagoras parses out the administrative content of each one and placed it into an <<*AskOptions*>> or <<*AskOptional*>> block, as appropriate, at the top of the document. An additional 'instruction' block is added containing usage information.

3. Review the prompts that were created. Modify the questions as appropriate to make the appropriate response as unambiguous as possible.
4. Save the document.

 **Clause Sets:** If you assemble documents using [Clause Sets](#)⁴⁰⁶, the actual <<*Options*>>, <<*Optional*>> and <<*Repeat*>> blocks may not be present in the 'raw' document, and therefore are not going to be picked up by the tool discussed above. Here are the steps to follow in that instance:

- create a 'complete' document which contains all possible options and optional text blocks.

It is not important at this stage that the resulting document contains more text than would ever be used. The idea here is to have a document that contains all of your <<*Options*>>, <<*Optional*>> and <<*Repeat*>> elements.

Before you create this 'huge' document, you should turn 'off' processing so that the <<*Options/Optional/Repeat*>> blocks are not automatically processed.)

- Run the automatic <<*Ask*>> command creator described above.
- Edit the <<*Ask*>> commands to your satisfaction.
- Copy and paste the entire block of <<*Ask*>> commands into the first clause that you use when assembling via Clause Sets. (Typically, the 'first clause' is the same throughout all documents in the topic, but not necessarily.
- If some <<*Ask* >> commands seem inappropriate to a particular Clause Set, just delete it (for example, when a clause set creates a Will for a family with no minor children, delete <<*Ask*>> commands referencing a !Guardian!).

Limitations:

1. The source <<*Options/Optional*>> block must each contain a !GroupName!. Reason: the <<*Ask*>> command must have a !group! reference in order to link with <<*Options/Optional*>> blocks further below in the document.
2. The <<*Options/Optional*>> must either contain 'prompt' statements/questions or, if no such statements, actual text options whose combined length is less than 150 characters. (If longer than 150, an <<*Ask*>> block will still be created, but you will be prompted to create manually the actual prompts that will be asked of the end user.)

25.5.7.2 Logic Assistant

Pathagoras provides a Logic Assistant to help you to construct the equations necessary to perform logical operations within your document.

Activate the Logic Assistant from the Pathagoras Features Menu | Authoring Tools. Click the Logic Assistant element. The following screen will appear:

Scan the screen for a second. Note its "If" this value exists, then "perform or set" this value. Here are the steps to complete the assistant.

1. Click one of the radio buttons at the top of the screen. These options reflect the various types of Ask elements that can exist in a document (Options/Optional/Repeats). In our example, we choose Optional because "Are there children?" is presented as an "AskOptional" in the Ask Table.

2. As soon as you click on the Ask element (1), the left and right panels are populated with choices that you can make. At the left, select the element against which you want to apply your logic equation. Here, we want to apply logic against the element 'Children.' That is, "IF there are Children . . . do something."
3. Continuing with the "IF" part of the equation, select the appropriate value of the element that is being compared.
4. While this '4' is not a step, we point out that Pathagoras is beginning to construct the logic formula for you. Use this as a reference point if you become confused.
5. Look at the screen below. In the middle panel, make a selection as to which logic step is

to be first presented. The choices are simple: to display the 'Ask' associated with the target value, or to set a specific value (True or False, or one of the choices) for the target. The check box reflect which appears first.

Document Logic Assistant

Select the type of prompt that will provide the 'test' value when answered:

- ☐ Options (multiple choice)
- ☒ Optional (keep or discard)
- ☐ Repeats

This form is designed to assist you in creating logical IF/THEN type equations with your <<*Options/Optional/Repeats*>> blocks.

Click "type" of block (options, optional or repeats) from the choices at the

If this Optional block term is selected:

(Choose from below !Groups!)

- Children
- Minors
- Withdrawal Rights
- Special Use

Perform/Set (check = Show First)

- ☒ ... the Ask associated with selection at the right, and
- ☐ ... the value(s) selected below, right

... on this value.

(Choose from below !Groups!)

- Married
- Sex of Client
- Children
- Minors
- Withdrawal Rights

is:

- ☐ 'True' or 'Yes'
- ☒ 'False' or 'No'

Sample:

<<*If*!Children!= "Yes"; <<*AskOptional*!Minors!Minor children?*>>, !minors!= "No">>

<<*If*!Children!= "True"; <<*AskOptional*!Minors!Are there minor children?*>>, !Minors!= "False">>

Transfer to Asks Table

- From the right panel, select the element that will be acted upon (here we have selected 'Minors') based on the value of the selection at the left. Once we select the interrelated elements from the left and right panels,
- Choose the value of the element from the left panel that is to be measured. In the case of Optional blocks, the only values are "True" or "False". (For "Options" blocks, the values can be any combination of the actual choices.) All two possible actions are shown.

IMPORTANT: Check "where you are" frequently. Always keep an eye to the formula bar (8) that changes every time you make a new selection. None of the instructions above will make sense in the vacuum of this page, but when you see the formula change as you press buttons, we assure you that the Logic Assistant will communicate what it needs. Plus, read the elements of the form. Note the left to right layout and 'descending' displays that make it easier to visualize the formula you are creating.

- Check the formula bar to make sure that the "If THIS, Then THIS, Else THIS" is what you desire.
- Make changes as needed. When done, click the Transfer button.

When you click the Transfer button, Pathagoras hunts down the element in the Ask table that is associated with the target (in this case "Minors") and replaces it with the formula. So the Ask table shown at the top of this page will now look like this (check out the 4th row):

<<*AskOptions(radio)*!Married!Married/Not Married*>>
<<*AskOptions(radio)*!Sex of Client!Male Client/Female Client*>>
<<*AskOptional*!Children!Are there children?*>>
<<*If*!Children!= "True", <<*AskOptional*!Minors!Are there minor children?*>>, !Minors!="False">>
<<*AskOptional*!Withdrawal Rights!Exercise Annual Withdrawal Rights.*>>
<<*AskOptional*!Special Use!Special Use Valuations*>>

Here is the logic equation found in the 4th row, in 'English':
 "If there are children, then ask the question 'Are there minor children?'
 Otherwise, set the value of the !Minors! group to 'False' ."

25.5.8 Limitations

Size Limitations:

The physical size of the Interview form that Pathagoras can display on the editing screen is a function of two distinct factors

1. **Program limitations:** A potential of up to six Options block can be listed along the left side of the overlay. A potential of 10 "Optional" text blocks can be listed along the right side of the overlay.

Note: If the document contains more that six <<*Options*>> blocks or more than ten <<*Optional*>> blocks, only the first six <<*Options*>> blocks (subject to 'height' limitations discussed below) and the first ten <<*Optional*>> blocks will be displayed. The remaining blocks will be carried over to a second 'run' of the Interview routine.

2. **Screen Resolution:** The resolution of your computer (and the number of pixels that the

'screen height' parameter allows) may further limit the display of <<*Options*>> blocks along the left side of the Interview form.

Note: If the amount of space needed by Pathagoras to display all <<*Options*>> blocks exceeds the 'height' on the editing screen to hold the Interview questions, the undisplayed questions will be carried over to a second 'run' of the Interview routine. This will happen automatically.

Nesting Limitations:

Pathagoras allows nesting of <<*Options*>> within <<*Optional*>> blocks, <<*Optional*>> within <<*Options*>> blocks, and any combination thereof. And there is no limit to the level of nestings that are allowed, although beyond 2 levels may become somewhat confusing.

However, for nesting to work, each <<*Options*>> block within a nest must have either a !groupname! or a series of short questions (or prompts) associated with it. (There actually is an exception to this, as actual usage will reveal. If the <<*Options*>> text is short, Pathagoras will process it with no problems. Nevertheless, 'best practices' suggest that you attach a !groupname! or prompts.

See also:

[<<*Options/Optional*>> text blocks](#)  141

[!GroupNames!](#)  151

[Nesting](#)  148

25.5.9 Interview Documents

This is not really an interview in the sense discussed above. (In the above examples, the interview questions are placed at the top of a source document, and values are assigned and processed.) Here, we are creating essentially a document titled "Interview" with a single question that results in the selection of an appropriate second document. This is shown as an alternative to the more elaborate Interview process described in this section. It is yet another way in the Pathagoras arsenal to guide an end-user to an ultimate 'destination.'

Here is an example:

1. Create a document called "Will Interview." For this example, it will contain a single Options text block consisting of one multi-part question, with a series of document references enclosed within "<<" and ">>" brackets being the 'answers. However, it can contain anything you want.

The individual references are to documents that contain text appropriate to the response to the options presented. Here is what "Will Interview" might contain.

```
<<*Options*!Family Structure!Married with one or more children/Married, no children/Single
(divorced or never married), with one or more children/Single (divorced or never married),
no children*>>Married with Children>>/>>Married no Children>>/>>Single with Children>>/
>>Single No Children>>>>
```

The 'administrative section (the text in blue) of the options block above provides 'questions' that will be presented to the end user when that user calls up the "Will Interview" document. The remainder of the text block provides the references to the document that will be called based on the selection made. The "<<" and ">>" marks surrounding the document names indicates that you intend a document (or glossary clause) by that name to be called. (Without those boundary markers, you would get just text.) (Please note that the coloring you see is for illustrative purposes.

Colors are not required in the actual template or clause.

Of course, there must actually be documents or glossary terms called "Married with Children.doc", "Married no Children.doc", etc. which respectively provide the text appropriate for "Married with one or more children", "Married, no Children" etc., wills.

2. The answer to the options posed in the 'first' document will result in a call to an appropriate 'second' document. This second document can contain 'terminal' questions (the full Interview, as it were) or can lead to additional documents in the Interview sequence. There is no limit to the nestings and references that can be provided and performed.

Notes:

- Regular <<*Options*>> and <<*Optional*>> [construction rules](#)¹⁴¹ apply.
- The references can be to a complete document/template (as suggested in the above example) but they also can be to [Clause Sets](#)⁴⁰⁶.
- If the reference is composed using the [prefix/suffix naming convention](#)⁴²⁰ (as shown in the example), or if the referenced clauses are stored in the Super Folder or Super Glossary, Pathagoras can quickly find desired clause without more in the clause name. See [Search Order Rules](#)⁹⁵.
- If the reference is not in prefix/suffix style, and the referenced clauses are not in the Super Folder or Super Glossary, the target clause either (1) must be found in the same folder as the initial Interview form (and this is likely to be true) or (2) must be fully qualify (with drive and folder designators).

Don't confuse the Interview (a top of document entity)

with the <<*Options* and *Optional*>> blocks (a document body element)!

The interview has its own structure rules. The interview is entire optional and most users (currently) don't use it because it does involve a bit of programming. The Interview asks questions and, as asked and answered and values set, they disappear. The interview does not, is not intended to, and cannot be used to provide substantive text within its boundaries. Just a question (converted into an interview form).

"Asks" and "Ifs" are uniquely Interview functions. They cannot be used within Options and Optional text blocks. Options and Optional text blocks are document body elements

So, the following is NOT a proper Optional block:

```
<<[*!b!="Yes",[Borrower Name], [**a] [Borrower State of Organization] [borrower entity type],!b!="No"[Guarantor Name], [**a] [Guarantor State of Organization] [guarantor entity type]>>
```

But THIS IS FINE:

```
<<*Optional*!b![Borrower Name], [**a] [Borrower State of Organization] [borrower entity type]>>
```

OR (if you want to provide a 'negative' value, add a slash)

```
<<*Optional*!b![Borrower Name], [**a] [Borrower State of Organization] [borrower entity type]/[Guarantor Name], [**a] [Guarantor State of Organization] [guarantor entity type]>>
```

Note that the Optional block structure is very simple and straight-forward. It contains no 'logic' (If / then) language. It is very straight forward. It tells you right off the blocks

'purpose' (i.e, that it is "Optional" or "Options" text and then it presents the text. If you want to set a group value, or to link the text block with a !group! value that was earlier set, simply type the group name immediately after the Optional or Options call. Then provide the text results. Nothing more. No 'ifs' required (or allowed). See above sample.

25.5.10 Saving <<*Ask* . . . records

Starting with version 2012.1, after you have made choices on the 'Wizard' screen and click the Next button, Pathagoras will ask if you want to save the selections. If you think you will reuse these choices (either in case you have to remake the current document or if you are likely to create new documents with the same values), say 'Yes'. Provide a record name when prompted. (This is similar to the Instant Database used to record personal values that will replace variables. Pathagoras will record your choices for later use and then proceed to implement your selections.)

If a second or subsequent screen of the Wizard appears, Pathagoras will automatically record the choices in the same file named in the initial screen. Pathagoras will make all changes to your document (making Options choices, deleting optional text you elected not to keep, repeating text blocks and incrementing variables as appropriate, etc.) as reflected in your selections.

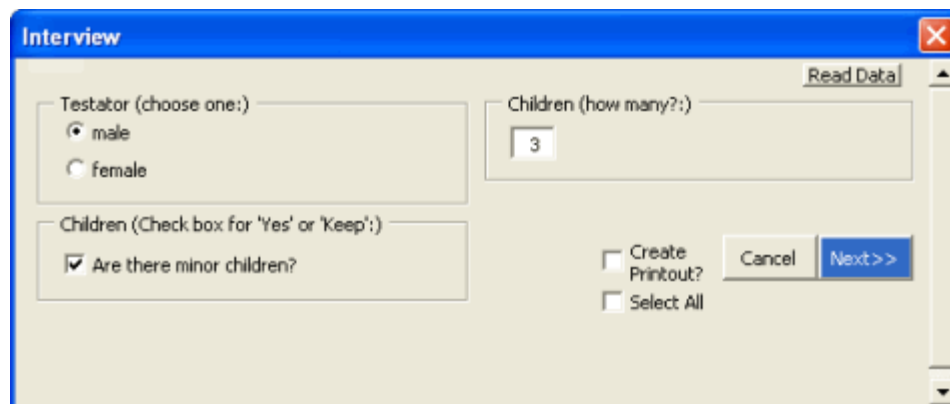
25.5.11 Recalling <<*Ask* . . . record

When you have assemble a document which contains an <<*Ask* . . . >> table, the Wizards screen will display. If there are <<*Ask* . . . records that you have previously saved (see previous page) a button will appear in the extreme upper right corner labeled "Read Data". If you wish to recall a record instead of manually making each selection, click that button.

From the screen that next displays, select the desired record. Then, press the 'Go' button on the overlay screen.to continue.



Pathagoras automatically will select the radio buttons and checkboxes and fill in the 'repeat' values in accord with the saved data. (The example given, with only 3 choices on screen, hardly exemplifies the value of this features. But when there are multiple screens and dozens of choices, this can be a tremendous time saver.)



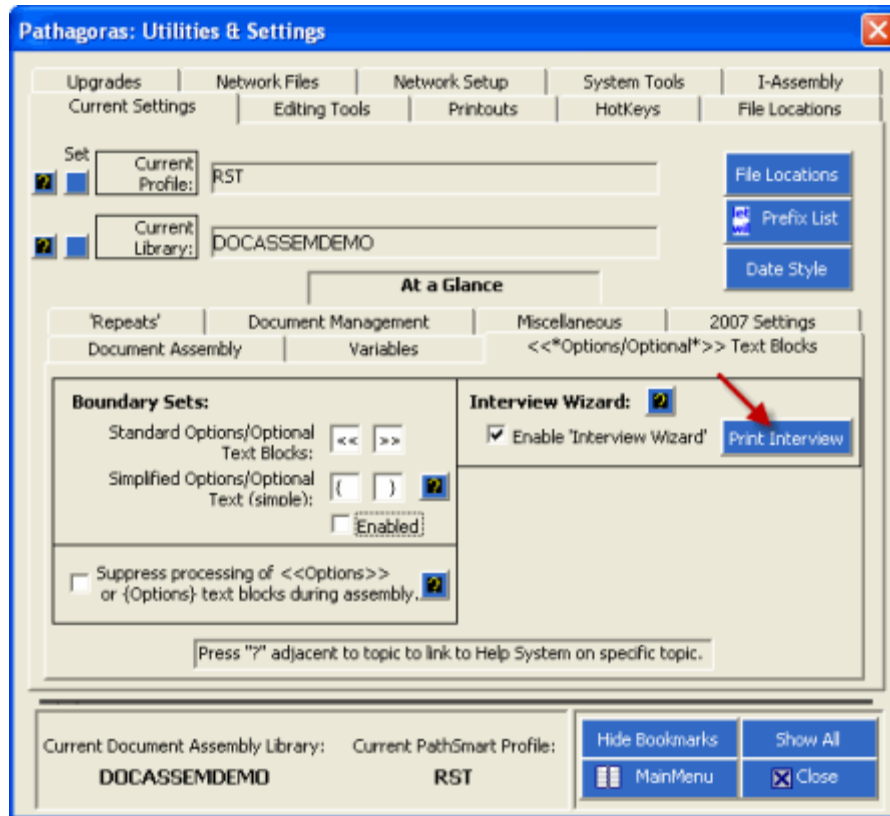
Make any changes as may be appropriate. Pathagoras will ask you if you want to update the existing record or create a new record. This second (create new record) feature allows you to record a 'standard' or 'universal' set of answers perhaps named 'Standard' --or anything you wish-- so that Pathagoras will automatically select the 'typical' values for you. You then can modify the selections as appropriate and save the new, more personalized, record for the specific client or customer. (This feature is similar to the 'mask' function of Instant Database records.)

25.5.12 Print Interview

Generate a printout of the various <<*Options*>> and <<*Optional*>> "questions":

Select the <<*Options/Optional*>> tab

Note: The printout is actually a Word document. Freely edit the document All questions will print, including nested questions, in the order in which they appear in the document.



25.5.13 'Case' Logic

'Case' logic is a more complex extension of the If/Then/Else logic discussed earlier. The term 'Case' derives from the sentence "In CASE a certain value(s) is (are) THIS, then do this"

Case logic is especially useful for comparing multiple values, multiple combinations of values, and for determining values in a cascading fashion.

- If/Then/Else statements contain basically two result sections: the True part and the False part. It is 'linear' in structure, reading from left to right, "If this value is X, then do this, else do this."
- 'Case' logic allows many result sections. It is 'stacked' in structure, and reads from top to bottom. Example (read the commas as "then"):

Case

Combination 1, A action
Combination 2, B action
Combination 3, C action
etc.

- 'Combination' above refers to the choices made in a preceding <<*AskOptions*>> display. The combinations refer to the positions of the selections made. (Caveats: If the display was from an <<*AskOptions(radio)*...>> call, only one position, not combinations of positions, will be 'analyzed'. Ditto If the display was from an <<*AskOptional*. . .>> call.)
- When multiple values are involved, multiple considerations must be handled. Here's

how we do it.

- The answer combination is always converted to a positional number. So, if five items are in the list of choices, and the user selects the second and fifth item, Pathagoras will 'see' the answer numerically as 2 and 5.
 - If all the choices must be present in order to trigger the case event (but others can be as well), just type the qualifying choices, separating multiple choices with commas.
 - If only the exact choices listed must be present to trigger the case event (and none others), enclose the choices within curly braces. (The choices themselves must still be separated by commas.)
 - If any one of the listed choices will trigger the case event, separate the choices with '/OR'.
- When the first true value is reached, the line is processed and the routine ends.
 - The Case list can be endless.

Below is an example that demonstrates how the initial selection of items (in this example, salad fixings) can lead to a cascading of subsequent selections including the salad name and the utensils needed to consume the salad.

You can copy and paste this example into a document and process it. Pathagoras will ask you the first question dealing with the salad's "ingredients". If the user selects a predesignated combination of ingredients, Pathagoras will automatically return the common name of the salad and the utensil commonly used to consume it. Otherwise, the AskOptions questions (the final entry in the Case list) are presented.

```
<<*AskOptions*!Ingredients!Celery sticks/Julienne Carrot Strips/Raisins/Apples/
Bananas/Cream Cheese/Peanut Butter/Crackers*>>
```

```
<<*Case*
```

```
!Ingredients!= "1,6",!Salad!="1"
```

```
!Ingredients!= "2,3",!Salad!="2"
```

```
!Ingredients!= "1,2",!Salad!="3"
```

```
!Ingredients!= "3,4,5",!Salad!="4"
```

```
<<*AskOptions*!Salad!Stuffed Celery/Carrot and Raisin Salad/Sliced Veggies/Fruit
Salad/Unappetizing mismash*>>
```

```
>>
```

```
<<*Case*
```

```
!Salad!= "1",!Utensils!="4"
```

```
!Salad!= "2",!Utensils!="2"
```

```
!Salad!= "3",!Utensils!="4"
```

```
!Salad!= "4",!Utensils!="4"
```

```
>>
```

(The top part of this document is the 'Ask Table'. The below is the 'actual' document.
Below are six 'primary ingredients' for making a side dish of some sorts.

INGREDIENTS:

```
<<*Options*!Ingredients!*Celery sticks, / Julienne Carrot Strips, /Raisins, /Apples, /
Bananas, /Cream Cheese/Peanut Butter/Crackers*>>
```


Objective:

We want Pathagoras to select the specific salad (below) if a known combination is selected from among the ingredients above. So, if a pre-determined combination of ingredients is selected, the proper return will be made. Otherwise, Pathagoras will present a question allowing you to manually select the answer. Then, once the kind of appetizer is determined, Pathagoras will determine (or provide a list for you to select) the utensil to be used to consume the appetizer.

The type of SALAD made from the above ingredients is:

```
<<*Options*!Salad!*Stuffed Celery/Carrot and Raisin Salad/Sliced Veggies/Fruit Salad/
Unappetizing mismash>>
```

For the above side dish, we will need the following UTENSILS:

```
<<*Options*!Utensils!*Knife and Fingers/Fork/Spoon/Fingers>>
```

A reminder: the answers that Pathagoras is analyzing in the Case list are positional. "1,6" refer to the first and sixth positions in the ingredients list (celery and cream cheese). For <<*AskOptional*>> commands, a 'yes' answer is assigned a '1' value, a 'no' answer a '0' value.

Creating a Case Logic List.

1. Display the Logic Wizard (previous section).
2. Make all selections as before, but check the box in the middle of the screen that reads "Create Case Logic Block"
3. After you have created your first Case Logic item, click the Transfer button. The initial formula will be copied into the clipboard and the screen will close. Navigate to the position in the Ask Table where you want the Case Block to appear and click Paste (Ctrl-V).
4. Note: You can only create the *structure* for the case block, which will consist of the Case title, one (and only one) Case entry PLUS the Ask associated with the primary element selected at the right (this Ask must be the last entry, as it is the 'default item' if no prior Cases are true when document processing occurs).
5. You can (and should) add as many additional 'tests' as you wish, following the pattern shown in the transferred Case Logic Block.
6. If all selections must be met, enclose the selections within { curly braces }. Otherwise, the default 'one or more of the choices must match' will apply.'

Here is another, perhaps more 'practical' example. The clauses <<wil100>>, etc are clauses that shipped with the program. Therefore, you can copy and paste this example into a document and actually process it.

Objective:

To automatically select specific clauses based on choices made by the user from an Options or AskOptions block.

In the below example, a variety of family structure elements are presented. These choices are presented to the user once document processing starts. Once the user makes selection, the choices are evaluated by subsequent 'Case' blocks. (That is if any are provided. Case is optional. Pathagoras can evaluate the AskOptions answer just fine for other Options blocks further in the document Case is used for its ability to set a second or third value that will be used elsewhere.. This is called 'conditional branching'.)

If a pre-determined combination of choices has been made (all selections based simple on position—1,2,3 etc in the Options list), the designated return of a clause will be made. If a pre-determined combination was not selected, the 'default' value provided in the 'Case' block will be chosen. (The default value is always the last choice provided. That is because once a line in the case block evaluates 'true', the Case block is exited. If the last line is reached, it is the only one left. In the example below, the last choice is the same as if "Married, w/Children" had been selected for !Family Structure! group and "No Minor Children" for the !Guardian!.group.)

More explanations provided below.

```

=====
<<*AskOptions(radio)*!Sex of Client!Male/Female*>>
<<*AskOptions(case)*!Family Structure!(1)Married/(2)No Children/(3)Child/(4)Children/(5)
Minor children*>>
<<*Case*
!Family Structure!= "1,2",!Clauses!="1"
!Family Structure!= "1,3",!Clauses!="2"
!Family Structure!= "1,4",!Clauses!="3"
!Family Structure!= "2",!Clauses!="4"
!Family Structure!= "3",!Clauses!="5"
!Family Structure!= "4",!Clauses!="6"
!Clauses!= "3"
>>
<<*Case*
!Family Structure!= "5",!Guardian!="True"
!Guardian!="False"
>>
=====

```

The above text is the 'administrative section' of the document. It comprises the 'Ask Table' and the Case processing section. The Ask Table asks the questions when the document is called up via the Document Assembly screen or a DropDown List (or if you 'fake' processing via Alt-P. The below is the 'actual' document.

If you want to suppress the "and" "or" "space" "none" connector options that typically appear below any AskOptions control, just append '(case)' after "AskOptions".

Here we add another element to the example, and that is the ability of Pathagoras to call in clauses set out in double angle brackets. All clauses to be inserted in the example below are actual clauses. They are part of the the DemoData clauses (Wills clauses) that shipped as part of the Pathagoras installation package. Of course, you can substitute real text in lieu of 'wil100'. etc.

Here, the 'Preamble' (wil100) will be inserted regardless of family structure (no 'if' or 'Options' or 'Optional' assigned to it.

```

=====
<<wil100>>
=====

```

Here, depending upon the selections made regarding family structure, one of the following clauses will

be inserted. The 'Family Structure' choice made at the outset set a 'Clauses' value. That value is used as a 'position' in the below list of Clauses.

```
=====
<<*Options*!Clauses!*<<wil110(m,c0)>>/<<wil110(m,c1)>>/<<wil110(m,c2+)>>/
<<wil110(s,c0)>>/<<wil110(s,c1)>>/<<wil110(s,c2+)>>>>
```

=====

Now, throw in a few more standard clauses:

=====

```
<<wil120>>
<<wil130>>
```

=====

If 'minor child' was selected in the Family Structure ask, the Guardian clause will be inserted. Otherwise it will be omitted. (This is controlled by the 5th element of the family structure questions as analyzed by the second "Case" block above.)

=====

```
<<*Optional*!Guardian!*I hereby name [Guardian] to be the guardian of my minor
children.>>
```

=====

Now, finish up with a few standard closing elements and the signature block based on the answer to the opening Ask prompt 'Sex of Client'.

=====

```
<<wil180>>
<<wil190>>
<<*Options*!Sex of client!*<<wil300m>>/<<wil300f>>>>
```

25.5.14 Requirements

The Interview form can be created dynamically. There is no design required.

However, the Interview is generated based on the existence of <<*Options*>> and <<*Optional*>> text blocks in the document which control the selection of text by the user. Therefore,

- You must have <<*Options*>> and <<*Optional*>> text blocks in your document. See [this section of the Manual](#)¹⁴¹ for information on how to construct (and simple it is to construct) these powerful text blocks.
- Each <<*Options*>> and <<*Optional*>> text block must contain a !GroupName!. It is this !GroupName! that is checked to determine if a particular <<*Options*>> and <<*Optional*>> text block will be moved into the Interview form. (If there is no !GroupName!, the text block will be processed, but after the Interview is created and processed.)
- Regular <<*Options*>> and <<*Optional*>> construction rules apply. If you are not sure if your document complies, run the "[Structure Checker](#)"¹⁷³. Pathagoras will fix things for you automatically. Use the [Options Block Creation Assistant](#)¹⁵⁵ to help you correctly form the text blocks.

25.5.15 Interview Examples

Enter topic text here.

25.5.15.1 Simple Interview Examples

Here are a few examples of <<*Ask*>> prompts. (The first line is the Ask. The other lines are the <<*Options*>> or <<*Optional*>> text blocks against which the !group! value is applied.) You can copy any (or all) of the examples to your editing screen and see how the elements work together.

```
<<*AskOptions*!Customer!male/female/more than one*>>
Memo to JRT:
Send this package to <<*Options*!Customer!him/her/them*>>.
<<*Options*!Customer!He is/She is/They are*>> going to sign it send it on to
<<*Options*!Customer!his/her/their*>> attorney.
*****

<<*AskOptional*!Children!Will there be young children attending?*>>
We are having a party. Date: [Date of Party]
Location: [Location of Party]
<<*Optional*!Children!*Your children are invited.*>>
Please bring a covered dish and your favorite beverages.
<<*Optional*!Children!*Make sure that your children bring their favorite toys.*>>
*****
```

Here is an example that illustrates the point about a 'less than obvious answer' *Options* block. If the *Ask* prompt were not present, an end user who is unfamiliar with the periodic table likely would not be able to process the document with confidence:

```
<<*AskOptions*!Metal!Gold/Iron/Carbon/Mercury*>>
Here is a great experiment, boys and girls, that you can try at home:
Pour 10 ounces of finely ground <<*Options*!Metal!*Au/Fe/C/Hg*>> into a bowl.
Add a 16 ounces of H2O and 3 jelly beans. Mix well. Heat in oven at 400 degrees for 5
minutes. Let cool 22 minutes.
Out will come a shiny new <<*Options*!Metal!*Ring/Car/Diamond/Thermometer*>>!
```

(My lawyer made me do this: Please for goodness sake, do **not** try this at home (or anywhere else)!! It won't work. Besides, 400 degrees is really hot!)

This example illustrates *cascading logic* in an Estate Planning setting. It captures

- whether there are children in the mix and if so,
 - ❖ how many children there are
 - ❖ If just 1, whether the child is a 'son' or 'daughter' ("To my child" in a Will sounds so 'stiff' whereas 'to my children' always sounds 'good'), and
 - ❖ whether any child is still a minor (so that perhaps a Guardian can be appointed).

(Note: This is not the only approach, so don't hesitate to be creative and design an even more

efficient cascade.":

```
<<*AskOptions(radio)*!HasChildren!Client has children/Client has no children*>>
<<*If*!HasChildren!="1",<<*AskRepeat*!NumChildren!Number of
Children*>>,!NumChildren!="0(#)">>
<<*If*!NumChildren!="1",<<*AskOptions(radio)*!ChildGender!Son/Daughter*>>,>>
<<*If*!NumChildren!>"0",<<*AskOptions(radio)*!AllAdults!All children adults/Minor child(ren)
*>>,!AllAdults!="1(2)">>
<<*If*!NumChildren!>"0" and !AllAdults!="2",!Guardian!="True",!Guardian!="False">>
```

This example addresses what a manufacturing company might face in processing special orders. There are extra manufacturing charges and extra shipping charges associated with custom orders. Here the ~ (not equal) is illustrated.

```
<<*AskOptions(radio)*!Size!Small/Medium/Large/Custom*>>
<<*AskOptions(radio)*!Color!Red/Blue/Green/Custom*>>
<<*If*!Size!="Custom"AND !Color!="Custom",<<*AskOptions*!ExtraCharge!$50/$60/$70*>>,>>
<<*If*!Size!="Custom"AND !Color! ~
"Custom",<<*AskOptions*!ExtraCharge!$100/$150/$200*>>,>>
<<*If*!Size!="Custom OR !Color!="Custom",<<*AskOptions(radio)*!Shipping!$20/$30/$40*>>,>>
```

25.5.15.2 A Whimsical Example

The previous sample has a useful application. This one does not, but perhaps by its very non-standard nature, it will make more sense to some:

--Copy Below--

The below gets or sets values via the AskTable.

Ask Block Text	Before processing	After processing
<<*AskRepeat* !NumCdn!*How many children?>>	Repeat block created in normal fashion, with 'Repeat' designator and groupname followed by the question. Let's say "2".	The AskRepeat line will disappear, but "2" has been assigned to !NumCdn!
<<*AskRepeat* !NumCats!*How many cats?>>	Repeat block created in normal fashion, with 'Repeat' designator and groupname followed by the question. Let's say "4"	The AskRepeat line will disappear, but "4" has been assigned to !NumCats!
<<*Set*!NumDogs!=3(#) *>>	The values in parentheses relate important information. If a '#', it says the value is a raw number. NumDogs is set to the raw value of 3 (dogs);	The Set line will disappear, but "3" is assigned to !NumDogs!
<<*If*!NumCdn! +!NumDogs!>"5",!Crazy House!="1(2)",!CrazyHo use!="2(2)">>	If a number, it conveys that the Group is an 'Options' (vs. Optional) block and the total number of choices. Here, !CrazyHouse! is an Options (not Optional) block and there are 2 choices provided in the document.) If the value set were "True" or	Pathagoras performs the math. Since !NumCdn! plus !NumDogs! is more than 5, !CrazyHouse! is set to "1" (or the first) of 2 choices.

	"False" instead of a number, the block would be "Optional".	
<pre><<*Set* !NumCritters!=!NumDogs !+!NumCats!>></pre>	The above sets another value based on the input and/or set value.	Pathagoras performs the math and sets a value. Since !NumDogs! plus !NumCats! equals 6, so !NumCritters! is set to "6".

The below is body text. The results of the AskTable are carried into the body text.

The names of my child(ren) is/are <<*Repeat(and) *!NumCdn!*[Child]>>	This illustrates Pathagoras' 'repeat action' using the raw value of the AskRepeat for NumCdn.	The names of my child(ren) is/are [Child@1],[Child@1] and [Child@1]
The names of my dog(s) is/are <<*Repeat*!NumDogs!*[Dog]>>.	This illustrates the 'repeat action' using the raw value of the Set function for NumDogs.)	The names of my dog(s) is/are [Dog@1], [Dog@2], [Dog@3].
I have <<*Options*!NumCdn!*no children/one child/two or more children>>.	This illustrate the 'Options' action which transforms the Repeat value of NumCdn: '0'=first option; '1'=second option; '2+'= third option.)	I have two or more children.
<pre><<*Options*!CrazyHouse!*T his is a crazy house with <<!numcdn!>> {!NumCdn!children/ORchild/ ORchildren} and <<!numdogs!>> dogs inside./This is a sane house with {!NumCdn!no children/ ORonly one child/ORonly <<!numcdn!>> children} and <<!numdogs!>> dogs inside.>></pre>	This illustrates (1) how Pathagoras can use a set variable (CrazyHouse); (2) insertion of raw Repeat and Set values via GroupNames; and (3) 'SimpleOptions' action which transforms the Repeat value of NumCdn: '0'=first option; '1'=second option; '2+'= third option.	This is a crazy house with two children and three dogs inside.
This house has <<!NUMCRITTERS!>> critters inside!	This illustrates returning just the 'raw' value. The number is spelled out and the case and emphasis is preserved.	The house has SIX critters inside.

--Copy above--

25.5.15.3 Another Whimsical Example

Pathagoras offers much more power and flexibility to choices made in response to <<*AskOptions*>> blocks.

In prior versions, the value of the !GroupName! was the numeric equivalent to the choices made. So if 9 choices were presented, and the first fifth and eighth items were checked, the !GroupName! was assigned the value of "1,5,8" Other document logic calculations could be made by reference to that value, but not others were possible.

Now, in addition to the base group being assigned a value (discussed above), the individual elements of the group are automatically assigned a True/False value (no additional work on your part). Calculations and evaluations anywhere in the document can be done using those individual values.

An example works best. You can copy and paste the entire block of text to an editing screen. To process it, press <Alt-P>.

--Copy Below--

The below (left column) is a standard 'AskOptions' prompt.

Ask Block Text	Before processing	After processing
<<*AskOptions*!Fruit!Apples/Bananas/Cherries*>>	AskOptions block displays the groupname and menu choices you want to provide to end user.	This line will disappear once the Ask block has been answered. If Apples and Cherries are checked, the value of the group name 'Fruit' is set to '1,3'

Document body standard behavior:

Body Text	Before processing	After processing
<<*Options*!Fruit!I eat an apple a day./I like bananas in my cereal./I prefer a cherry on my sundaes.>>	Options block created in normal fashion, with 'Optional' designator and groupname followed by the body text.	Since Fruit = "1,3" (per above example), the second option is excised, leaving the first and third options in the document, leaving I eat an apple a day. I prefer a cherry on my sundaes.

NEW: Pathagoras automatically creates a 'true' or 'false' value for each element of AskOptions list which may be used in the document body.

Body Text	Before processing	After processing
<<*Optional*!Apples!*An apple a day keeps the doctor away.>>	Optional block created in normal fashion, with 'Optional' designator and GroupName followed by the body text.	'Apples' was checked in the AskOptions listing per our example above. A new GroupName called 'Apples' is established, and its value is set to 'True'. The text will remain.
<<*Optional*!bananas!*My doctors tells me that bananas are the perfect fruit.>>		Bananas was unchecked in the AskOptions listing. A new GroupName called 'Bananas' is established. Its value is 'False'. The text is deleted.
<<*Optional*!cherries!*Please top my [sundae/milkshake] with a cherry.>>		'Cherries' was checked in the AskOptions listing per our example above. A new GroupName called 'Cherries' is established. Its value is set to 'True' and it will remain.

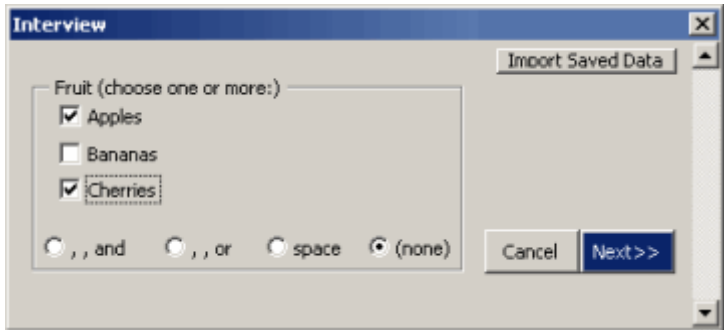
Optional can also accommodate 'True' and 'False' outcome.

To enable, just add a '/NEGOPT' (for negative optional) and some 'alternative' text.

If the !Group! value of the optional block is True, the text before the /NEGOPT is kept;
if false, the text after the /NEGOPT is kept.

Body Text	Before processing	After processing
<<*Optional*!apples!*An apple a day keeps the doctor away./ NEGOPTThose wormy apples didn't keep my doctor away.>>	Optional block created in normal fashion. However, a 'false' value added. A slash separates the 'True' value from the 'False'. (Coloring, italics and underlining for illustration only.)	'Apples' was checked in the AskOptions listing per our example above. A new GroupName called 'Apples' is established, and its value was set to 'True', so the 'true' side of the Optional block will remain.
<<*Optional*!Bananas!*M y doctors tells me that bananas are the perfect fruit./NEGOPTYes, I have no bananas.>>		'Bananas' was unchecked in the AskOptions listing per our example above. A new GroupName called 'Bananas' was established, and its value was set to 'False', and the 'false' side of the Optional block will remain.
<<*Optional*!Cherries!*P lease top my [sundae/ milkshake] with a cherry./NEGOPTI don't want any cherries. They have pits.>>		'Cherries' was checked in the AskOptions listing per our example above. A new GroupName called 'Cherries' is established, and its value was set to 'True'. The 'true' side of the Optional block remains.

--Copy above--



This is what the selection menu would look like when the above text is initially processed.

Not only will Pathagoras assign a value of "1,3" to the GroupName !Fruit!, but it will automatically assign "True" to !Apples! and to !Cherries! and "False" to !Bananas!.

You can then use these values anywhere in the document in the same way any !groupname! is used.

25.5.15.4 A 'Legal' Example

Below is a short sample document incorporating a few of the above features. It results in a set of reciprocal Husband and Wife Wills. It is based on the same text presented in the

'Dynamic Creation of Variables' section of the Manual, with more of the document logic features of Pathagoras added. Each of the major features shown in blue links to the section of this Manual that defines and further illustrates them.

Features in use:

[<<*AskOptions*>>](#)⁴⁹⁹ and [!Groups!](#)¹⁵¹

Notes: AskOptions can be created [automatically](#)⁵⁰⁹ from the existing <<*Options*. . .>> blocks and placed in an attractive table at the top of the document. Click the above link to see how. Here, the prompt is used to determine which of the Will is to be created first. The !GroupName! allows the answer at the top to 'affect' the Options block further down in the document.

[<<*If*>>](#)⁵⁰⁶

Notes: Here, the prompt used to determine if the "Minor Children" question should even be asked. If there are children, no sense in asking if there are minor children.

[<<*AskRepeats*>>](#)³⁸⁴ with series connectors and !Groups!

Note that once the number of children is determined, it is used in 2 distinct ways. One way is to repeat the raw value an appropriate number of times against the variables '[Child@Name]' and '[Child@DOB]'. When attached to an <<*Options*>> block, the 'other way' kicks in. Rather than repeating a variable, it selects a phrase depending upon where there are '0' children, '1' child or '2 or more' children.

You can copy the sample text in the box below by placing your cursor inside the box, pressing Ctrl-A (to select 'all' and then Ctrl-C (for 'copy'). Then paste the text into a document (Ctrl-V) (this one has no mnemonic equivalent) . Process the page (Alt-P) to see how it all comes together.

```
<<*AskOptions*!Whose Will!First Is Husband's First Is Wife's*>>
<<*AskOptional*!Children!Are there children?*>>
<<*If*!Children!="Yes",<<*AskRepeat*!NumChildren!How many
children*>>,!minors!="No">>
<<*If*!Children!="Yes",<<*AskOptional*!Minors!Are there minor
children*>>,!minors!="No">>
```

Last Will and Testament of

<<*Options*!Whose Will!*[HUSBAND NAME]/[WIFE NAME]>>

I, <<*Options*!Whose Will!*[Husband Name]/[Wife Name]>>, being of sound mind, make this document my Last Will and Testament.

1. I give all of my property and estate to my <<*Options*!Whose Will!*wife/husband>>, <<*Options*!Whose Will!*[Wife Name]/[Husband Name]>>

2. <<*Optional*!Children!*If my <<*Options*!Whose Will!*wife/husband>>, does not survive me, I give all of my property to my <<*Options*!NumChildren!*child/children <<*Repeat(and)*!NumChildren!*[Child@Name], born [Child@DOB]>>.>>>>

3. I appoint my <<*Options*!Whose Will!*wife/husband>>, <<*Options*!Whose Will!*[Wife Name]/[Husband Name]>> to be my Personal

Representative.

4. <<*Optional*!Minors!*If at the time of my death I have any minor children, I appoint [Names of Guardians] to be the Guardians of such minor children.>>

<<*Options*!Whose Will!*[Husband
Name]/[Wife Name]>>

====Page Break=====

Last Will and Testament of <<*Options*!Whose Will!*[WIFE NAME]/[HUSBAND NAME]>>

I, <<*Options*!Whose Will!*[Wife Name]/[Husband Name]>>, being of sound mind, make this document my Last Will and Testament.

1. I give all of my property and estate to my <<*Options*!Whose Will!*husband/wife>>, <<*Options*!Whose Will!*[Husband Name]/[Wife Name]>>.

2. <<*Optional*!Children!*If my <<*Options*!Whose Will!*husband/wife>>, does not survive me, I give all of my property to my <<*Options*!NumChildren!*child/children <<*Repeat(and)*!NumChildren!*[Child@Name], born [Child@DOB]>>. >>>>

3. I appoint my <<*Options*!Whose Will!*husband/wife>>, <<*Options*!Whose Will!*[Husband Name]/[Wife Name]>> to be my Personal Representative.

4. <<*Optional*!Minors!*If at the time of my death I have any minor children, I appoint [Names of Guardians] to be the Guardians of such minor children.>>

<<*Options*!Whose Will!*[Wife Name]/
[Husband Name]>>

See these other examples:

[Legal pleadings: Captions using <<*AskRepeat* . . . >>](#)  399.

[Mixing 'Standard' and 'Simplified' Options Blocks](#)  170

['Repeat' Blocks](#)  382

25.5.15.5 Dissecting an Ask Table

Below is a sample of an Ask table that contains a wide variable of elements. It is a 'perfect' sample for discussion.

1	<<*AskOptions(radio)*!ClientSex!Client = Male/Client = Female*>>
2	<<*AskOptions(radio)*!Married!Married/Single*>>
3	<<*If*!Married!="1"and !ClientSex!="1",!SpType!="2(2)",!SpType!="1(2)">>
4	<<*AskRepeat*!NumCdn!Number of Children*>>

```

5 <<*If*!NumCdn!="0",<<*AskOptional*!AllAdults!All children adults?*>>>>
6 <<*AskOptions(radio)*!ResTrust!Residual Trust/No Residual Trust*>>
7 <<*AskOptions(radio)*!T=Trustee!Testator is Trustee/Testator not Trustee*>>
8 <<*AskOptions(radio)*!TrSign!Trust signed today/Trust signed earlier*>>
9 <<*If*!ResTrust!="1",!Trustee!="True",!Trustee!="False">>
10 <<*AskOptions(radio)*!POA!Powers of Appointment Not Exercised/Powers of Appointment
    Exercised*>>
11 <<*AskRepeat*!ExecNum![Executor]>>
12 <<*AskOptions(radio)*!corpExec!Corporate Executor/No Corporate Executor*>>
13 <<*AskOptional*!MultiSuccessorExecs!Multiple successor executors?*>>
14 <<*If*!AllAdults!="True",!Minors!="False",!Minors!="True">>
15 <<*If*!NumCdn!>"0" and !AllAdults!="True",!Guardian!="True",!Guardian!="False">>

```

1. Simply determines the sex of the client. Not the (radio) attribute. That insures that, when the Interview is presented, then user can select only one of the two choices.
2. Determines marital status and assigns it to the groupname !Married?!
3. The (*If*) line compares two previous values to determine a third. The determined value is clear in the underlying document – !Sp Type! will be either 'Husband' or 'Wife'. However, in logic equations such as this, we must speak in terms of position, not absolute value. Here is how the line is evaluated:
 - If the value assigned to !Married! is the position 1 answer (in this case "Married") AND if the !ClientSex! answer is "Male" (also the position 1 answer in its group), then the !SpType! will be the second choice ('Wife').
 - If the !ClientSex! is '2', then !SpType! will become '1' (or 'Husband')
 - The number in parentheses is optional. It is used to indicate how many choices that the !groupname! presents in the main document. In this case 2. While optional, it is much preferred.
4. Line 4 asks for the number of repeating variable given the groupname !NumCdn! (number of children)
5. If the number assigned to the groupname !NumCdn! is 0, then it assigns the 'true' value to 'nothing', but this 'nothing' still needs to be stated. That is done simple with a 'blank' comma. That way, Pathagoras can know where the 'false' value starts. (Note: this could also be re-written with a different comparator: <<*If*!NumCdn!>"0",<<*AskOptional*!AllAdults!All children adults?*>>>>,>>)
6. Line 15 contains a multiple comparisons, connected by 'and'. It is just another way of stating lines 5 and 14.

Notes:

- a. GroupNames can be anything. They should help you and the end user (if the end user might see the Ask Table) to determine the purpose of the group. The question mark is not a mandatory element (The same groupname could have been !Married! or !Client is Married! or anything else).
- b. Typically, the short the GroupName the better, but only because it takes up less space. So !ST! arguably would be better than !SpouseType!. The drawback is that a later editor may not understand what !ST! stood for and may become frustrated. So, in the example, we settled on !SpType!
- c. The placement of the various lines within the Ask block are not critical, so long as precedent values have been assigned before a dependent value is determined. So line 3 above could actually be anywhere, including line 11, so long as the values that that line needs to complete the evaluation (in this case, 1 and 2) precede it.

- d. The evaluation process uses classic Boolean logic structure. If 'this', then 'that', else 'other that'. The astute reader will observe that the logic equation in 3 is imperfect. If the parties are not married and ClientSex = 1 (or 2), the equation will evaluate to the 'false' side, SpType = 1. However, as the document is structured, the value is acceptable because in actuality, it doesn't matter. All 'married' references are processed out of the document when the 'Client not married' selection was made, and it simply does not matter if the client is not married to a Husband or not married to a Wife. The same pertains to line 5. If there are no children, it doesn't matter if selection assigned to 'AllChildren' is Adults.
- e. As Pathagoras reads down the Ask table, it will stop at and evaluate all 'If' lines. If the precedent values have already been assigned, it will proceed forward. If not, it will stop, and present another Interview screen for completion of the required variables.
- f. If the value assigned to an undeclared GroupName is 'True' or 'False', the type of block in the underlying document will always be "<<*Optional*>>", with 2 and only 2 possible values: True or False (or Keep or Delete). Otherwise, it will be assigned as an "<<*Options*>>" block, with the number of elements as stated in the parenthesis, if provided.



An <<*Ask. . *>> table comprises plain Word text. You can edit it to suit your office's needs. Just highlight, cut and paste until you have the order of questions you wish to present to the end user.

25.5.15.6 Dissecting another Ask Table

Let's assume that, after you have run the Ask Creator, you have this table at the top of your document:

<<*AskOptions(radio)*!Married!Married/Not Married*>>
<<*AskOptions(radio)*!Sex of Client!Male Client/Female Client*>>
<<*AskOptional*!Children!Are there children?*>>
<<*AskOptional*!Minors!Are there minor children?*>>
<<*AskOptional*!Withdrawal Rights!Exercise Annual Withdrawal Rights.*>>
<<*AskOptional*!Special Use!Special Use Valuations*>>

Look at the third group "!Children!" and the fourth group "!Minors!". Logic dictates that if there are no children, there can be no minor children. It would be better if the answer to that fourth question were automatically set to 'No' (and, of course, the question not presented) if that situation pertains.

That is done by adding an "If" element to the equation. The structure of an 'If' element is fairly simple. You must be at least a bit familiar with what is called 'If/Then logic', which is simply a standard structure for writing a logic equation of the nature we are using here. Simply put, an 'If/Then' equation would state: "If this situation exists, then do this; else do this."

Once you frame in your head the logic steps, it's easy to move on to the actual equation. In Pathagoras, the "If" statement logic would be "If" the values assigned to "!Children!" is "True", then "Ask" "Are there Minors?"; otherwise, don't ask. Here is the formula in 'code' that Pathagoras can recognize:

```
<<*If*!Children!="True",<<*AskOptional*!Minors!Are there minor children*>>,>>
```

Syntax: <<*If* and the final >> are the boundary markers to the equation.
 !Children! is the GroupName of the value we are testing.
 ="True" is the testing value for the GroupName
 , concludes the If part of the equation
 <<*AskOptional*!Minors!Are there minor children*>> is the 'True' part, and simple tell
 the program to Ask the question.
 , concludes the 'Then' part of the equation
 (Since nothing follows the 'Then' part, the Else part is simply 'null.' But in the below
 example, the value of !Minors is affirmatively set to "False")

Let's say you want to affirmatively set !Minors! to 'False' when there are no children. Just
 rewrite the above as follows:

<<*If*!Children!="True",<<*AskOptional*!Minors!Are there minor
 children*>>,!Minors!="False">>

The Pathagoras System

Document Management

Part



26 Document Management

Document management deals with efficient storage and retrieval of documents. Pathagoras offers a very powerful, but uniquely simple, module for document management.

Pathagoras' document management module comes in two parts. The first is called '**PathSmart**' (handling document retrieval) and the other '**SaveSmart**' (handling one-click document storage).

PathSmart eliminates time consuming 'navigation' of folders (clicking up, down and across the Windows folder tree) to open.

SaveSmart (PathSmart's 'sister') eliminates the navigation otherwise required when you wish to save a document into a particular folder.

PathSmart/SaveSmart accomplishes its 'magic' by letting you assign numbers and nicknames to folders that you currently use to store documents. When you initiate a PathSmart/SaveSmart call, Pathagoras will take you immediately to the desired folder with only a reference to the number.

Pathsmart/SaveSmart does not capture your system, nor does it force you into a single time-consuming method of saving and retrieving documents. It augments what Word offers, leaving all Word tools intact.

26.1 Definitions

A **SmartPath** is a standard Word/Windows folder that has been 'mapped' (assigned) to a number and nickname. Pathagoras uses the number to directly and instantly access the folder.

➔ The module is called PathSmart, but each line on the screen (i.e., each pointer or mapping) is called a SmartPath

A PathSmart **profile** is a collection of up to 12 *SmartPaths*. Profiles are generally associated with a user or with an office function. Pathagoras allows an unlimited number of profiles and therefore the user can map an unlimited number of paths.

- A profile is typically given the name of a user or the name of an office section. Think of profiles as being highly personalized collections of the folders that the user would frequently access for non-document assembly purposes. Retrieving existing documents is the most common activity.
- The name of the *current* profile displays on the PathSmart screen at the right in the drop down list so named. Change the profile by clicking the list and selecting another.
- Add a new profile via the Settings button.

A '**SuperSmartPath**' is a regular *SmartPath* with one additional, but very powerful, feature. A **SuperSmartPath** allows you instant access not only to the files in the folder to where it points, but to all sub-folders beneath it.

See also: [Setting Smart Paths](#)  536

[Setting SuperSmartPaths](#)  537

[Compare Document Management with Document Assembly](#) ²⁴

26.2 PathSmart Screen

To display the PathSmart (document management) screen click the Document Assembly icon (it is the first button from the left in the Pathagoras toolbar – it is the ‘fast runner’ depicting speed.)



Figure 1. The Pathagoras Toolbar

The PathSmart screen will appear:

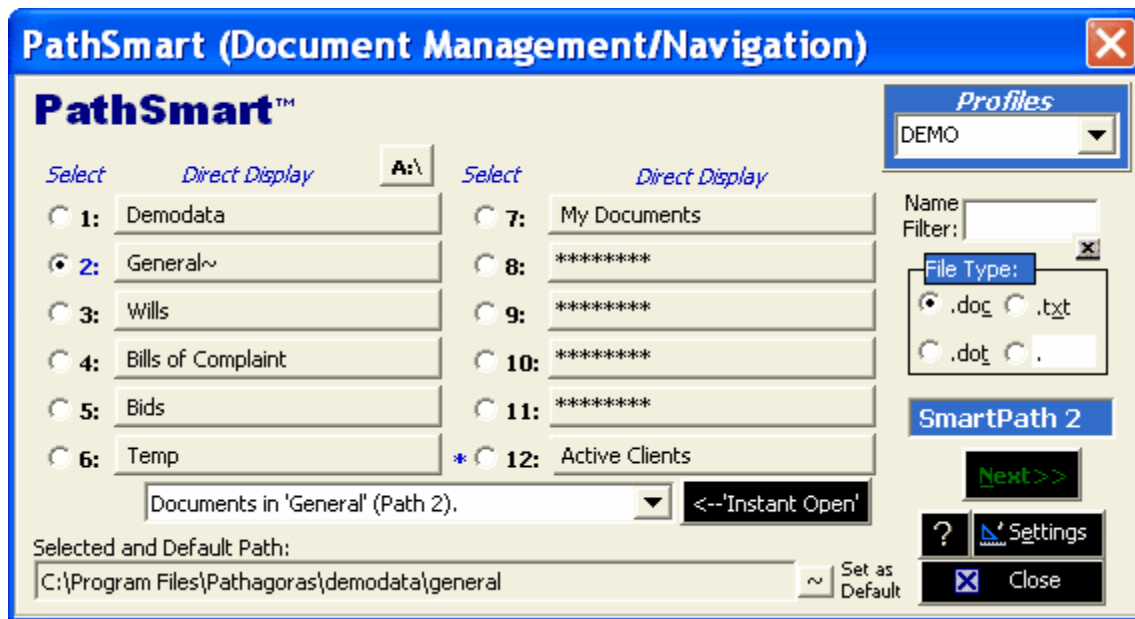


Figure 2. The PathSmart screen (initial view, default SmartPath 2 selected).
(Default SmartPath noted by '~')

SmartPath 2 (“General”) is the designated ‘default path.’ But let’s say that we don’t want to view the contents of SmartPath 2. We want 7. So we click the #7 radio button. (SmartPath7 is the only non-Pathagoras folder set in the demo. It is your default documents folder. Its value was derived from the Documents line in Tools>>Options>>File Locations. We called it ‘My Documents’ since that is the classic name for this folder, but it can be named anything you want. More on that later.)

Next, click the drop down list immediately adjacent to the Instant Open button to view the contents of the folder.

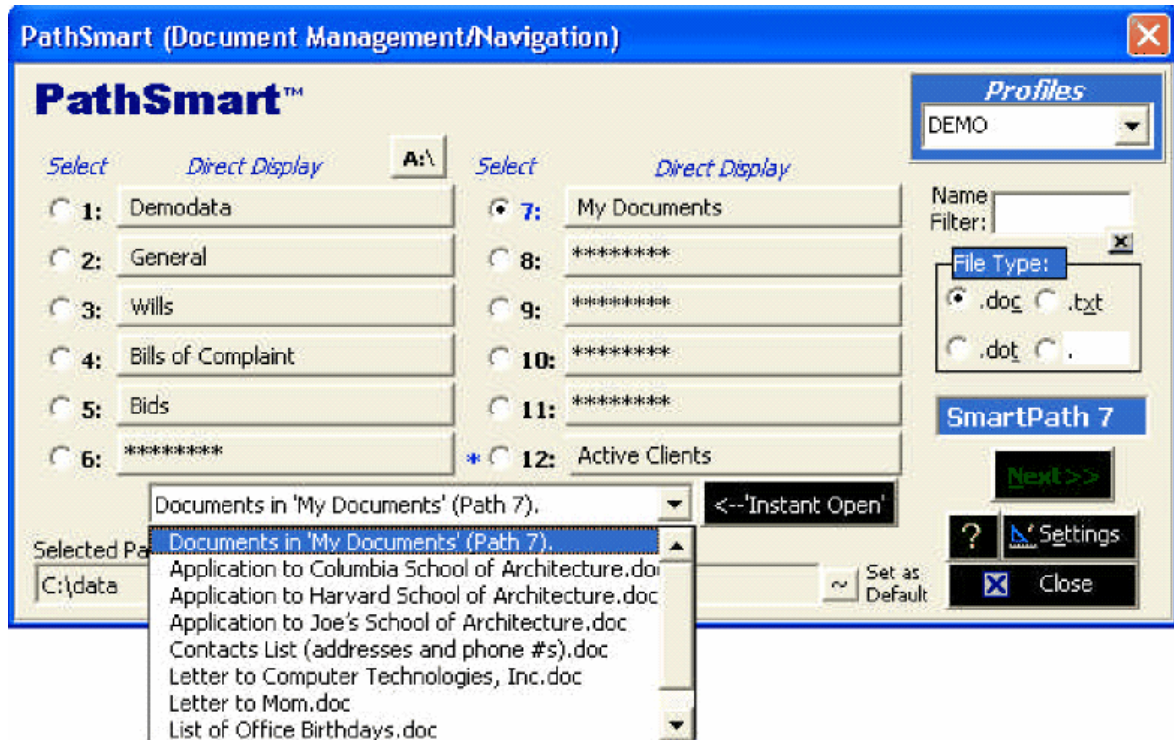


Figure 3. The PathSmart screen (after selecting Book 7 and dropping down the file list).

You can click on any item in the list to open up the document. Click the Instant Open button and, with absolutely no navigation, your document is opened. Or you can click another SmartPath option button and the contents of the selected SmartPath are displayed in the drop down list. See the Tip in the margin for more tricks on how to use this exclusive feature.

See Also:

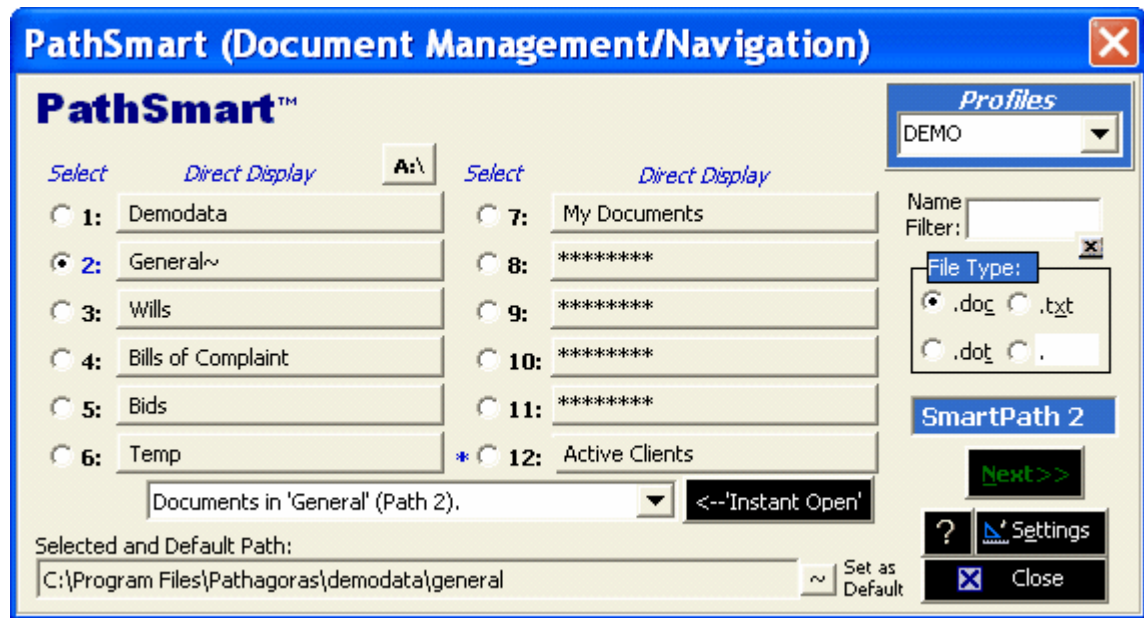
[Default Settings](#)  656

[Adding a SmartPath](#)  548

26.2.1 SmartPath

A *SmartPath* is a simple pointer to a specific folder. A SmartPath is identified by a number and a nickname. You can display the contents of a SmartPath in one of 3 ways.

1. **Press the option circle** to the left of the SmartPath nickname. This will display all of the documents in the selected SmartPath. If conditions were indicated in the Name Filter box, only the files meeting the conditions will be shown.



Note: When you select an option circle, the PathSmart screen does not close. If the document you are seeking is not in the selected *SmartPath*, click on another one. You can potentially scroll through every folder on your computer searching for a particular document without ever leaving this screen. And by using appropriate filter criteria, your search will be much faster than it could ever be by standard up, down and across navigation in basic Word.

2. **Press the name bar.** This will dismiss the PathSmart screen and display a standard Word File Open screen open to the selected *SmartPath*. If conditions were indicated in the Name Filter box, only the files meeting the conditions will be shown. By pre-setting the filter criteria, the display of filtered files is much faster than opening Word's File Open screen and then filtering for files.
3. **Mouselessly:** If you know the *SmartPath* number of the particular folder that you wish to open, simply type the *SmartPath* number onto your editing screen. Press the hot-key combination <Alt-G>. Word's File Open screen will instantly open to the proper folder. *No navigation required.*

See <Alt-G> for more 'tricks' using this hot-key (including filtered mouseless navigation, opening a specific document using just the *SmartPath* number and the document's name, and much more).

26.2.2 SuperSmartPaths

SuperSmartPaths

Do you keep documents generated for specific clients or customers in separate folders? Are all of those folders kept under an umbrella folder (called something like "Clients" or "Customers")? If so, Pathagoras' *SuperSmartPath* feature allows you to quickly access those documents in a way much faster than you could ever do in Word alone.

While retaining all other *SmartPath* features, when a numbered *SuperSmartPath* radio

button is clicked, Pathagoras displays the umbrella folder and all of the sub-folders beneath the assigned SmartPath. Here is the view (check out the bottom of the screen):

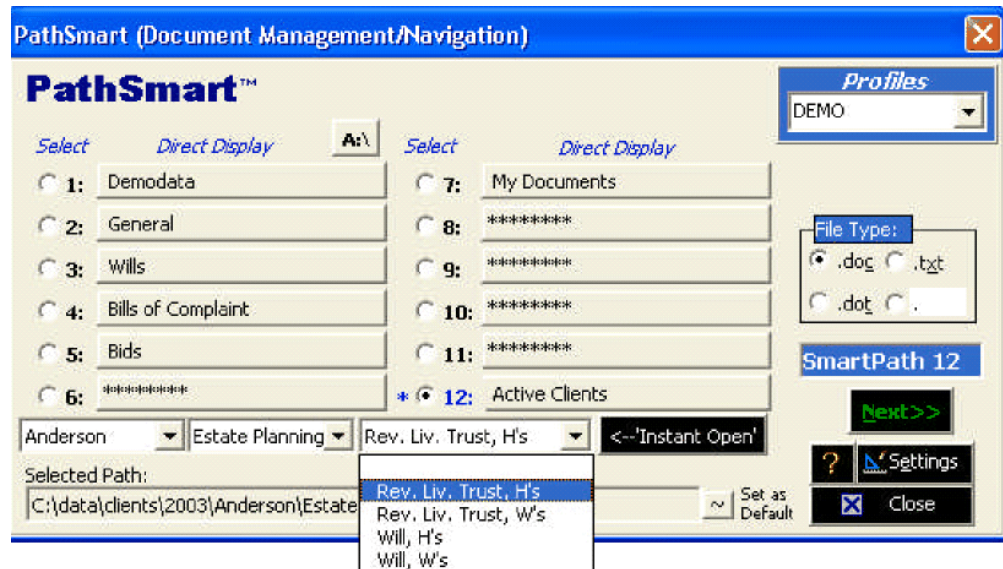


Figure 5. *SuperSmartPath* illustration.

Three new lists appear.

- The left-most list displays all sub-folders within the *SmartPath* (containing, perhaps, all clients or customers of the firm).
- The second list displays all sub-folders (if any) of the folder to the left.
- The third box displays the actual documents in the sub-folder.

To select a document to open, select the client/customer (*a*), the sub-folder (*b*) and the document (*c*). Then press the Instant Open button. The selected document is instantly opened onto your editing screen. (You can tell that you are working on the original document because its name appears in the upper left corner of the screen.)

A *SuperSmartPath* is the perfect solution for any ‘umbrella folder’ situation.

Creating a *SuperSmartPath*

- Assign a folder to a SmartPath using any technique previously discussed.
- Click the PathSmart <Settings> button and
- Check the *SuperSmartPath* box that corresponds with the *SmartPath*.

SmartPaths that have been turned into *SuperSmartPaths* are reflected as such by an “*” on the PathSmart screen.

26.2.3 Navigating the SmartPaths

It is possible to view all documents in the profile without leaving the PathSmart screen. Simply click each option button. The documents in the selected SmartPath will appear in the drop down list toward the bottom of the screen.

26.2.4 Range and Filter Controls

Pathagoras can filter, by name and by document type, for any document in any folder. When selected, only the documents that meet the filter criteria will display.

- If your folder contains many documents, you will find that the filtering occurs much more quickly than if you were to try the same thing in a regular Word File Open dialog.

'Name Filter': Use this to narrow the range of the documents that you want to show in the File List. Just type in the Name Filter the first few characters of the desired documents (e.g., "Abb" or perhaps "Let" if filtering for a document type). The file list will change instantly as you add or delete a character.

'File Type' filter: Select the type of file (by extension) that you want to display. Default is ".doc". The universal "*.*)" is a possible entry.

Note: The filter is sticky for the particular session. You can click on any or all of the radio buttons and scan every SmartPath for a particular document. If found, Pathagoras will so report. If not found, Pathagoras will report that no document matching the filter criteria exists.

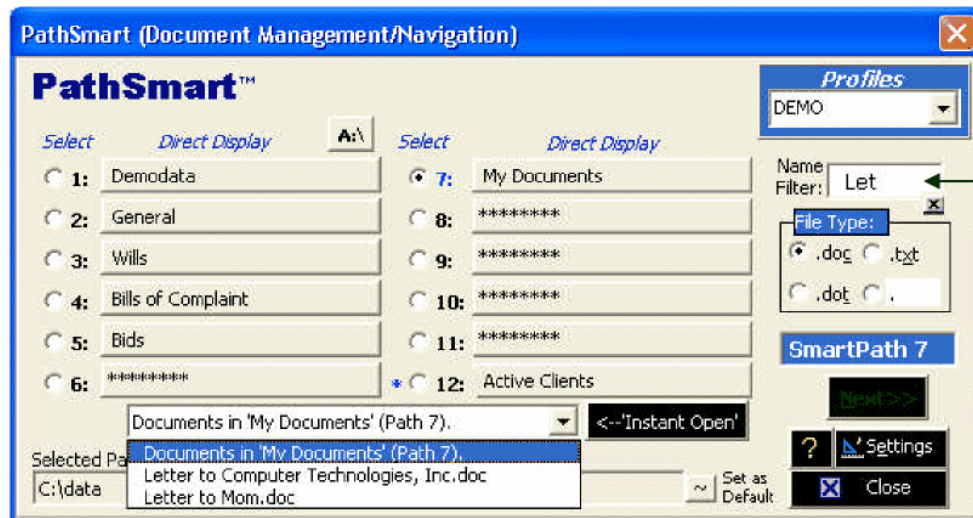


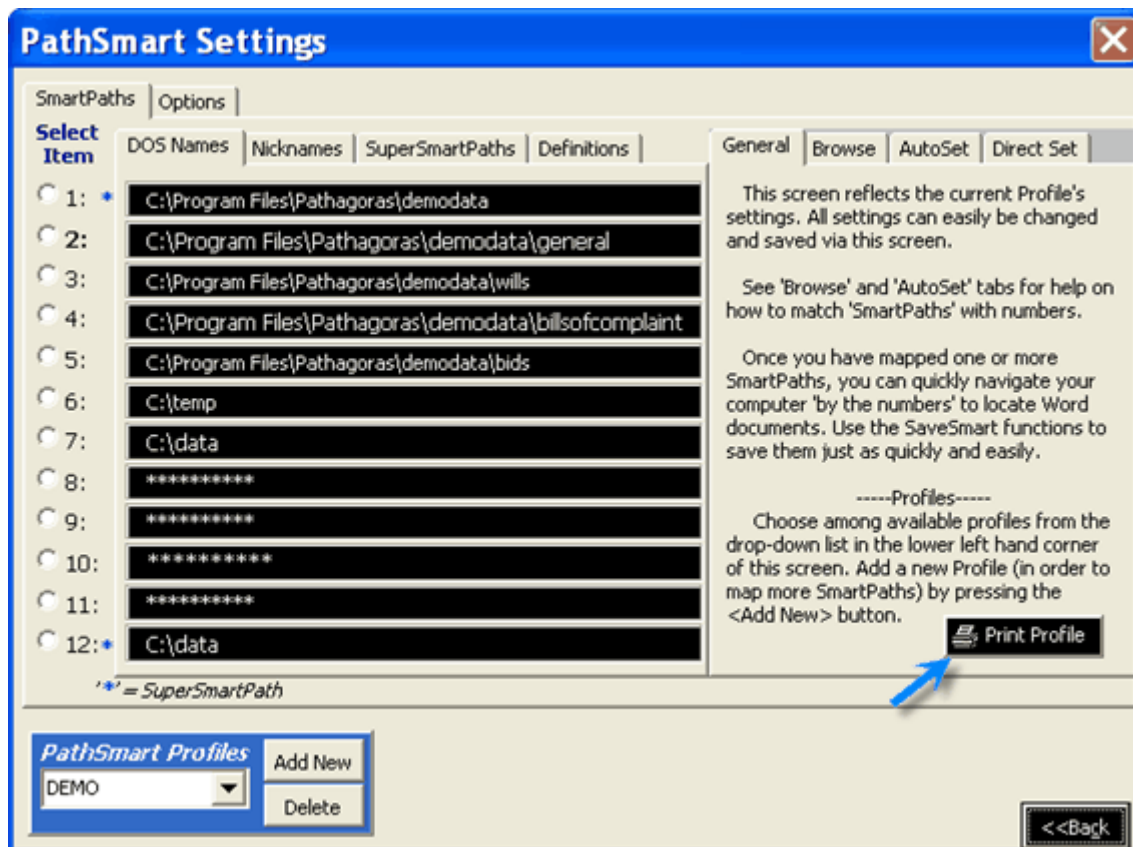
Figure 1. The filtered contents of My Documents (SmartPath #6). If your folders contain many documents, you will find that Pathagoras' filtering tools are much faster than Word's native filtering system.

To permanently select a 'File Type' filter, press Shift-along with the desired default.

26.2.5 Print Out SmartPaths

You can print out a handy list of the SmartPath assignments for a particular profile and post it for a quick reference

- (1) Display PathSmart
- (2) Click <Settings>
- (3) Click the <Print Profile> button.



Printout a list of SmartPath assignments via the PathSmart Settings screen.

The assignments will be sent to a blank document in either a 1 or 2 column format. Print the document and tape it to your monitor.

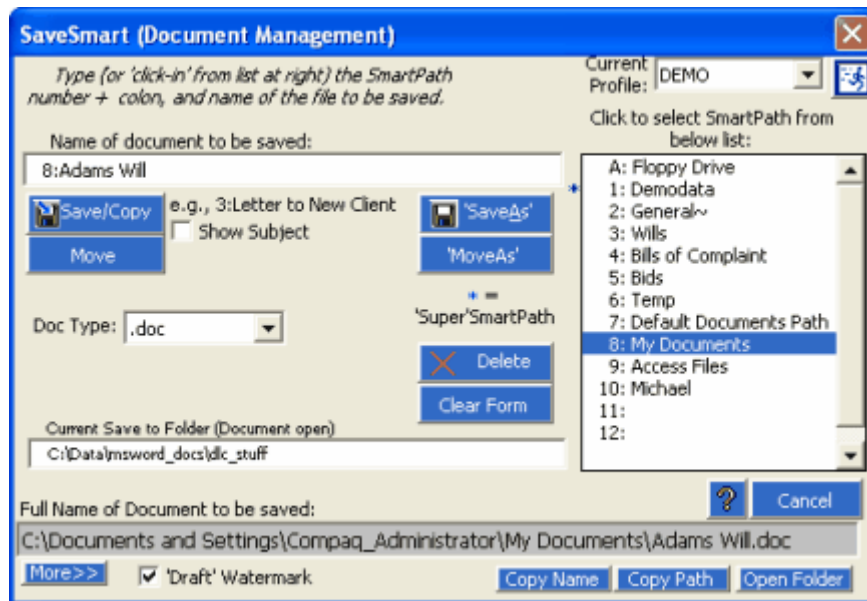
26.3 SaveSmart

After you have composed a document, you will need to save it to its proper place. You can use the standard Word 'Save' icon on the menu bar as you no doubt have done up until now. We highly recommend that you try SaveSmart. You can use SaveSmart to quickly save the on-screen document to the designated SmartPath without having to navigate to the location. The time you save will be worth the little bit of time needed to master the system.

To activate the SaveSmart features, simply click the 'disk' icon in the Pathagoras toolbar.



The Pathagoras toolbar



The SaveSmart Screen

Screen 'Geography' (details in subsequent sections):

- **Name of document to be saved:** This is the 'enter information' line where you provide (optionally) the SmartPath to where you want to save the document and the name of the document you wish to save. Note the changes to the grey textbox near the bottom of the screen as you type or change the SmartPath.
- **'Save/Copy'** is the default screen action. It saves the document to the location noted in the grey textbox near the bottom of the screen.
- **'Move'** means copy the document to a selected new location, and then delete the original. (Displays only when the active document is an existing, i.e., previously saved, document.)
- **'SaveAs'** displays the selected SmartPath folder, allowing you to navigate (or just look at the files that are already there to help you to decide upon a name).
- **'MoveAs'** displays the selected SmartSmartPath folder, allowing you to navigate (or just look at the files that are already there to help you to decide upon a name). After the Save, the original document is deleted. (Displays only when the active document is an existing document.)
- **'Delete'** allows you to delete the current document from your system. A warning is provided.
- The **panel** at the right displays the number and the assigned 'nicknames' of the SmartPaths in the profile shown at the top right of the screen. The bottom most entry ("13: Super Folder") will display once you have assigned such a folder. See Super Folder.
- **Current Save to Folder:** This line is informational. It reflects the folder into which the underlying document will be saved if you do not click in a SmartPath from the panel at the right. The box takes on a values dependent upon two scenarios

- If the open document is a previously saved document, the path to the document is the "Current Save To" path.
- If the current document is an unsaved, unnamed document (i.e., carries a temp name of Document1, 2, 3 etc.), the value reflects the default SmartPath assignment (as shown by a tilde '~' to the right of the name in the SmartPath list). If there is no default (i.e., no tilde), then the Current Save To path will be the default document path set in your Word settings.
- Full Name of Document to be saved. Hopefully this is self-explanatory.
- The **More>>** button provides additional tools for identifying and processing your documents. The More>> options will be discussed in greater detail in a [subsequent section](#)⁵⁴⁴.
- **Draft Watermark:** If checked, upon a Save, imprints a "Draft" watermark across the face of the first page of the document. If unchecked, removes the Draft watermark, if present.
- **Copy Name/Copy Path/Open Folder:** Oftentimes you want to know where the open document is saved. Oftentimes you want to open the folder of the current document, but Word's default file open folder is not that of the current document. Whenever you need to know this information, or go to the current document's folder, open SaveSmart and use one of the buttons.

26.3.1 Saving a Document

Saving to a 'standard' *SmartPath*

1. Type the name of your document in the text box toward the top/left of the screen
2. Click the <Save/Copy> button to save document (or press 'Enter' from your keyboard).
3. To save to a different folder, click folder from the list at the right or type in the SmartPath number, followed by a colon (':'), followed by the document name before clicking <Save/Copy>.

Your document will be saved in, or copied to, the folder shown in the 'Full Name' text box at the bottom of the screen.

➔ Note how the full document path shown at the bottom changes automatically when you name/rename your document and as you change SmartPaths.



Press <Shift><Save/Copy> and the document will be saved where desired, and then automatically closed.

Saving to a *SuperSmartPath*:

You may remember that a [SuperSmartPath](#)⁵³⁷ is an umbrella for all of the folders beneath it. When you begin a 'save' to a *SuperSmartPath*, the SaveSmart screen redraws itself to display 3 drop down lists.

- The first list displays all folders immediately under the umbrella folder. (This might be

all clients or customers by their names, if you created separate folders for each.)

- The second list reflects all sub-folders under the folder displayed in the first box.
- The third drop down list shows the specific documents that are either in the highlighted sub-subfolder (if such a folder exists) or in the active subfolder.

Select the appropriate folder from the first (and second) drop down list and provide a document name. Notice, as you are typing the name or changing the SmartPath, that the field at the bottom of the screen (Full Name of Document to be saved) changes as you type. Press <Save> and the document is instantly saved where you want it to be. Note: No navigation whatsoever!

Perhaps the document you have created is for a *new* client. With Word, you would have to navigate to the umbrella folder, create a new folder, create a sub-folder (if one is desired) and then name the document.

Pathagoras requires none of that. With Pathagoras, you simply type a name for the new folder (e.g., the client's name) at the top of the *first* list, type the name of the sub-folder (if any) at the top of the *second* list and provide the name of the document in the *third* field.

Pathagoras does all of the work in creating the required folders and naming the document.

Automatically. Instantly. No navigation.

26.3.2 Moving a Document

To Move a document using SaveSmart,

1. click the SaveSmart icon;
2. type the name of your document in the text box toward the top/left of the screen;
3. click the <Move> button to save the document.
4. To move the document to a different folder, click the folder from the list at the right or type in the SmartPath number, followed by a colon (':'), followed by the document name before clicking <Move>.

Your document will be saved in the folder shown in the 'Full Name' text box at the bottom of the screen. (Note how the DOS path changes automatically when you name/rename your document and as you change SmartPaths.)

Moving a document with SaveSmart is the same as <Save/Copy>, but after the document has been saved to its new location, the original version of the document is deleted from the original directory.

An unsaved document cannot be moved. Therefore, the <Move / Move As> buttons will be visible only if a previously saved document is displayed.

<Shift>-<Move> = the document will be moved where desired, and then automatically closed.

26.3.3 SaveAs/MoveAs

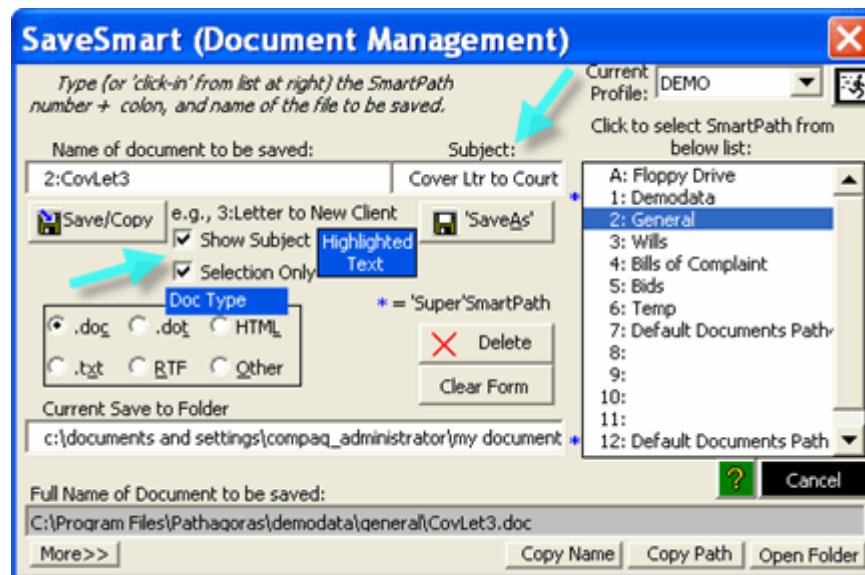
When the <SaveAs> button is clicked, Pathagoras will display a standard Word 'Save-As' dialog and return the 'Save' function to standard Word control. This allows you to manually navigate to a sub-folder (or parent folder, or any folder) before saving the document. You can also scan the list of documents already present in the folder.

<Move As> does the same as above. However, after you have copied the document to its new location, Pathagoras will erase the original document from its current location.

Document 'disassembly' using SmartSave: SaveAs

The SaveAs function under the SaveSmart screen allows you to highlight a section of text and to save it out as a new document in a more or less single step. It also allows you to

- highlight a portion of the entire document, and save just that portion as a new document.
- add Subject to the document directly from the SaveSmart screen.



See [Adding Clauses Manually](#)³¹¹ for more options and information.

26.3.4 Deleting a Document

<Delete>

If you know that you do not want to keep the currently displayed document, SaveSmart allows you to delete it while you are looking at it. Click the red X. A confirmatory “Are You Sure?” dialog will appear.

26.3.5 The More>> controls

Some very useful features are hidden within the SaveSmart screen that you may wish to explore. Click the More>> button and the screen changes to reveal a series of helpful tools.

Figure 2. More>> button reveals additional options.

Study the available options. The <Other Properties> button (found at the left side of the screen) displays the Summary Info screen. You can add the following information to the Properties section of the document.

26.3.6 Automatic Document Naming


The SaveSmart module contains functions whereby you can tell Pathagoras to automatically suggest a name for the document you are about to save.

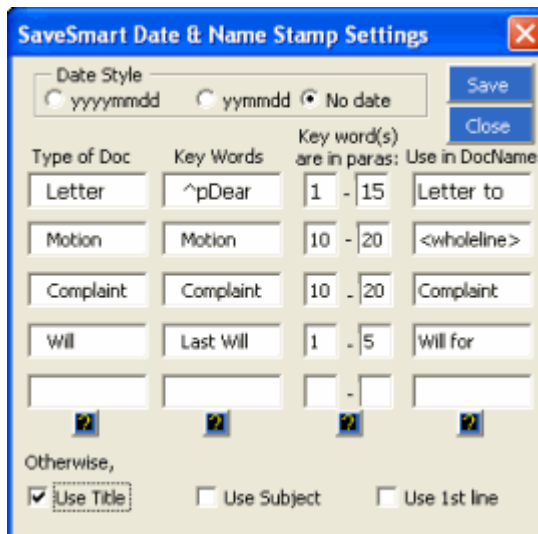
It operates on this very simple principle. If Pathagoras identifies key word or words within a designated range of lines at the beginning of a document, Pathagoras will name the document based on the key word(s) found.

For example, and depending upon the parameters you have set, if the word "Dear" is found between lines 5 and 10 of the document being saved, Pathagoras will 'think' that you are saving a letter and will offer to name the document "Letter to".

Note: You provide all of the parameters: the key word(s), the range of lines in the newly created document within which to search for the key word, and the proposed name for the document if the key word is found. All of these items are set and saved in the SaveSmart screen.

- (1) Click the SaveSmart icon and click the "More>>" button in the lower left quadrant of the screen.
- (2) Check the "Propose name of new document" box at the bottom of the augmented screen.
- (3) Click the button to the left of that checkbox to create your settings.
 - "?" help pop-ups are available at the bottom of each column to offer guidance as to what might go into each box. Here, you can set your key word(s), the lines within which Pathagoras should look for the key word(s) and the 'response' if the keyword(s) is found.
 - You can also append a date at the beginning of the proposed name by selecting one of the options at the top.

 You can set up to 5 sets of parameters. Here is a screen shot:



Type of Doc	Key Words	Key word(s) are in paras:	Use in DocName
Letter	^pDear	1 - 15	Letter to
Motion	Motion	10 - 20	<wholeline>
Complaint	Complaint	10 - 20	Complaint
Will	Last Will	1 - 5	Will for

Otherwise,
☒ Use Title ☐ Use Subject ☐ Use 1st line

To test your settings, open a new, blank, document and put the key word within the set range. Click SaveSmart and study the result. Modify the settings as appropriate.

Beneath the 'name based on content' section is a section titled "Otherwise". Here, Pathagoras provides you with additional choices for auto-naming your document. You may 'Use Title' or 'Use Subject' or 'Use 1st line', as desired. (Note, as the 'Otherwise' suggests, these latter choices are used only if none of the 'name based on content' choices are matched.)

26.3.7 Save to PowerPoint

You can save your work to a PowerPoint presentation via the SaveSmart screen. A particular (but simple) setup is required for the Word document in order to make the transition. Only "header" text is converted. "Normal" and other text will not transfer.

Each paragraph formatted with the Heading 1 style becomes the title of a new slide, each Heading 2 becomes the first level of text, and so on.

- (1) Click the SaveSmart icon and click the "More>>" button in the lower left quadrant of the screen.

Note: This feature is not something 'new' to Word. It is the same as the Send To PowerPoint feature found in Word's 'File' menu. We have added it to the SaveSmart screens so that it can be found more readily.

26.3.8 Save as Source Clauses



We have tried hard in this Help System to segregate the document management features (PathSmart and SaveSmart) from the Document Assembly features (everything else, including the Creation of Document Assembly 'Source Clauses.'

But SaveSmart should not be overlooked as a potential tool for building your source books. It actually is *one of the best tools* in the arsenal. Here is what you need to do:

1. Assign to a SmartPath one of your folders that currently houses document assembly clauses. We will call this the 'incomplete folder.'
2. Display a document which contains text that you want to add to the incomplete folder.

Note: This document can be anything. It can be a current document. It can be one you just downloaded from the Internet. It can be sample text that you received from attending a Continuing Education course. Think broadly here.

3. a. If you want to save the entire document, simply click the SaveSmart button and then, from the right panel, click the SmartPath number that represents the incomplete folder. Provide a name (and subject, if desired). Press the Save/Copy button and the text is saved as a new document in the designated folder.
- b. If you want to save just a portion of the document, highlight that portion first. Then click the SaveSmart button. (Pathagoras will know that you have highlighted text and will slightly change the face of the SaveSmart screen to accommodate.) Provide a name (and subject, if desired). Press the Save/Copy button and the text is saved as a new document in the designated folder.



Use SaveSmart to add documents/clauses to your [SuperFolder](#)⁹⁰. It is (if assigned) always SmartPath #13.

Note: You can use SaveSmart to save text only to a folder, not to a glossary.

26.4 Creating a New Profile

Creating a new profileA profile is a Document Management device that stores up to 12 SmartPaths.:

Click the runner icon (the first on the tool bar) to display the PathSmart screen.

Click the <Settings> button.

Click the <Add New> button.

Following the prompts, provide a name for your new Profile.

➔ A different naming ‘philosophy’ applies here. While in a document assembly setting, a library should be named for a universal audience (after all, many users might be tapping in to a library), a profile can be much more intimate. Its name should reflect the ‘personality’ of the operator, and its content should be the folders that the operator visits most frequently. We therefore recommend that a profile name be that of the user. E.g., “Mary” or “Robert.” As an exception, to the extent you want one of more ‘generic’ profiles to serve as the base for other profiles, consider giving that profile the name of the company.

You will be given a choice whether to keep the assignments of the current profile or to start ‘fresh.’ Select how much (if any) of the existing profile’s settings you wish to preserve.

26.5 Deleting a Profile

You can delete an entire profile (or simply remove a SmartPath) via the PathSmart Settings screen. Activate the screen. Display the profile you wish to delete (or from which you wish to clear or reassign a SmartPath).

If you wish to delete the entire Profile, click the Delete button in the lower left side of the Setting screen. A series of confirmation screens will appear to insure that you do not accidentally delete a Profile.

(If you wish only to remove a SmartPath, click on the option circle adjacent to the SmartPath's name. When you do so, a <Clear> button will appear. Click it to remove the SmartPath from the Profile. Press the <Save> button to confirm the action.)

NOTE: A Profile, and the SmartPaths it contains, are simply pointers to the folders that the SmartPaths represent. They are not the actual folders or the files in the folders themselves. Deleting a Profile or a SmartPath does not affect any substantive files.

26.6 Adding a SmartPath

Pathagoras provides two distinct ways to add a *SmartPath* to the current profile. If the *SmartPath* line is empty when you are viewing the PathSmart screen, you can Quick Set the SmartPath. [Click here for the instructions.](#)⁵⁴⁹

If you wish to replace an existing *SmartPath* with another, or switch *SmartPath* assignments, do so via the Settings screen. [Click here](#)⁵⁴⁹ for the steps.

26.6.1 Quick Set

Mapping an existing Word folder as a "SmartPath" via the PathSmart screen ("QuickSet").

Step 1: Click the PathSmart icon. The PathSmart screen will appear.

Step 2: Click on a blank line. Pathagoras will detect that the line is empty and will ask if you wish to assign a SmartPath to the line. Say 'Yes.'

Step 3: Following the prompts, navigate to the target folder that you want. The 'target folder' is the one containing the documents to which you want one-click (or mouseless) access. Select one item in the folder (it does not matter which one) and then click OK.

Step 4: When prompted, provide an appropriate nickname for the folder.

26.6.2 Via Settings Screen

Mapping a "SmartPath" via the PathSmart Settings screen.

1: Click the PathSmart icon.

2: Click the Settings button.

3: Click the radio circle next to the number of the SmartPath you want to set. Once you have clicked on a line, two new buttons will appear toward the bottom of the screen. One says 'Browse' and the other says 'AutoSet.' Click Browse

4: Following the prompts, navigate to the folder that you want to assign to the SmartPath. Select one item in the folder (it does not matter which one) and then click OK.

5: Following the prompts, assign a nickname to the folder.



If the document that is open on the screen happens (by design or by accident) to be from the folder to which you wish to set the SmartPath, use the AutoSet button in step 3. Step 4 will not be required.

26.6.3 Removing a SmartPath

You can clear a SmartPath (or delete an entire Profile) via the PathSmart Settings screen. Activate the screen. Display the profile from which you wish to clear or reassign a SmartPath. Click on the option circle adjacent to the SmartPath's name. When you do so, a <Clear> button will appear. Click it to remove the SmartPath from the library. Press the <Save> button to confirm the action.

→NOTE: A SmartPath is simply a pointer to the folder that the SmartPath represents. It is not the actual folder or the files in the folder itself. Deleting a SmartPath does not affect the actual files. Only the pointer in the SmartPath is deleted.

See also: [Deleting a Profile](#) 

26.7 Settings

You can set your *SmartPaths*, designate a *SmartPath* as a ***SuperSmartPath*** and set a variety of other settings via the Settings screen.

Please note that *SmartPath* settings are pointers to the folders that the numbers and the

nicknames represent. They are not the folders themselves. Deleting a *SmartPath* does not affect the actual files. Only the pointer in the profile is deleted, never any actual files.

See also: [Creating a New Profile](#)  ⁵⁴⁸

26.8 Mouseless Navigation

A primary reason for mapping a DOS path to a number (i.e., creating a *SmartPath*) is for the operator to be able to display that *SmartPath*, or a document within the *SmartPath*, 'mouselessly.' And the steps could not be easier.

To display the *SmartPath* (i.e., the folder itself):

1. Type the number of the desired *SmartPath* on any blank line of an editing screen.
2. Press <Alt-G>. For example:

8<Alt-G>

Instantly a standard Word 'FileOpen' dialog appears displaying the contents of the chosen *SmartPath*.

To open a document within a *SmartPath*:

1. Type the appropriate *SmartPath* number (1-12) on any blank line followed by a colon and the name of the document.
2. Press <Alt-G> for the *original* document or <Ctrl-Alt-G> to open a *copy* of the document. For example:

8:Letter to Mom <Alt-G>

See also [QuickLinks](#)

26.9 QuickLinks

Added with the release of Pathagoras 2016, **QuickLinks** is a variation of the Mouseless Navigation discussed in the immediately preceding section.

QuickLinks allows you to pair a short phrase with a folder that can be located anywhere on your system. Type the **QuickLink** to the screen and press <Alt-G> and a Word/Windows 'File Open' screen will be immediately opened.

How does it differ from PathSmart and its individual SmartPaths?

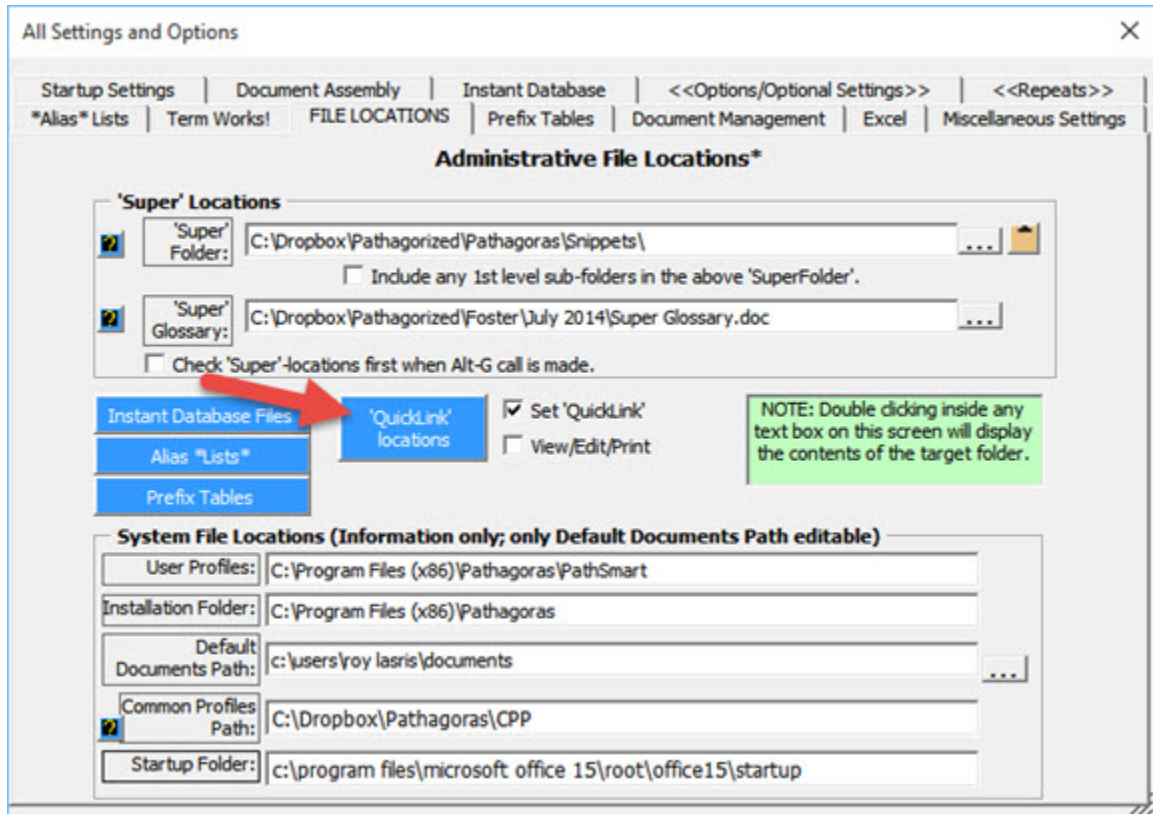
PathSmart and **QuickLinks** work in tandem, but with different access triggers. *PathSmart* works with numbers. **QuickLinks** work with names. If you know the number (1-12) associated with the *SmartPath*, type it, press <Alt-G> and Pathagoras will open the folder associated with the number. However, *SmartPath* is profile dependent, meaning that the number you type will pair with the *SmartPath* of the current *PathSmart* profile. A **QuickLink** will function independently of any profiles.

([Read more about PathSmart at this link](#)  ⁵³⁴.)

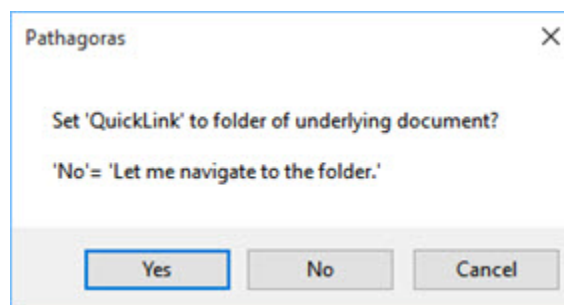
Creating a QuickLink:

1. Click the Pathagoras Setting screen and then click the File Locations button.
(Alternatively, you can click the Pathagoras Settings screen and then the **All Settings**

- button. Then click the File Locations tab.)
- Click the **QuickLink** button. (The 'Set QuickLink' box to the right will be checked by default.)



- The following screen will appear. Answer the prompt.



- If the underlying document is in fact in the folder you want to set, say 'Yes'. Otherwise, say no. then navigate to the folder to which you want to set a **QuickLink**.
- At the next prompt, give your **QuickLink** a name. Any term will do. It can be a single word or a meaningful phrase. That's all there is to it.

Using your QuickLink:

To Open Folders:

Simply type your **QuickLink** name or phrase on any line at the left margin. Press Alt-

G.

To Open Sub-folders:

If you happen to know a sub-folder that will get you closer to your target, you can type it (or the first few characters of it) as the QuickLink. *E.g.*, 'clients\ones'. Press <Alt-G>.

- If you typed the full sub-folder name, Pathagoras will display the Word/Window's File Open dialog with the sub-folder displayed.
- If you typed a partial name (e.g., clients\jo), Pathagoras will display a dialog displaying those folders that start with what you have typed (continuing with the example, it is as if you typed 'clients\jo*'). Select the desired sub- folder from the list.
- If you totally mistyped the name, Pathagoras will display the parent folder.

To Save files:

If you want to **SAVE** your on-screen document **TO** your **QuickLink** target folder, type the **QuickLink** name on any line at the left margin and press <Alt-S>. A normal Word/Windows Save As screen will open, displaying the target folder. Name your document in the normal fashion. (Don't worry about the **QuickLink** you typed. Pathagoras will erase it before you save the document.)

Editing QuickLinks

Check the 'View/Edit/Print' box and then the 'QuickLinks' button. Edit the **QuickLinks** in the 'NotePad' screen that will next appear. Just follow the same pattern: "*QL Name*", "*QL Path*" (two parts separated by a comma, enclosed in two sets of quotes)

Notes:

All of your QuickLinks will be valid for every user on the computer on which they were created. But for multi-license owners, you will have to set **QuickLinks** for each computer.

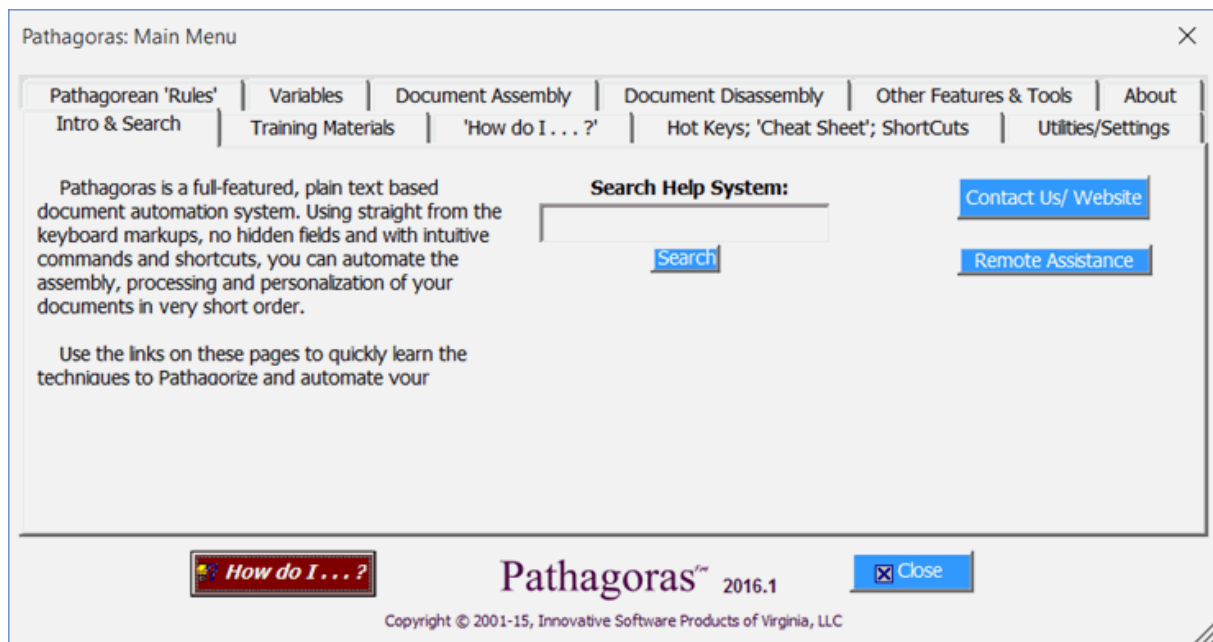
Even if the target folder assigned to your **QuickLink** is not precisely the one you want, it likely will be closer starting point than navigating from ground zero. Consider using **QuickLinks** for navigation beyond its specific target.

27 Main Menu

Pathagoras' Main Menu is the first item displayed in the Pathagoras DropDown Features list. You can trigger most functions from the Main Menu, but typically you will not once you have become familiar with the program. (Instead, you would press a hot-key combination, click the Document Assembly button, click on a DropDown List, etc.)

You likely will use the Main Menu as an access point to the Pathagoras Help Systems. Here are some other functions that are available:

- Perform a search of this Help system using the Search box provided in the 'front' page.
- Link to the 'How Do I . . . ?' system to learn how to perform certain Pathagoras actions.
- Link to, download and print, one or all of the Help manuals.
- Connect to a summary of the current Pathagoras' settings via the 'Settings button.
- Print out a helpful Reference Guide ('Cheat Sheet') which summarizes most of Pathagoras main functions and provides useful hints and tips for mastering the program.
- Contact Us.



The Pathagoras System

Network Operations

Part



28 Network Operations

A bit of preliminary information:

The single license version of Pathagoras is fully functional in any existing network environment. Regardless of the number of licenses owned (including one), you can

- point a document assembly *book* to any location that your computer can currently access, including across a network.
- point a *SmartPath* to any folder to which you now have the ability to navigate, including across a network.
- store all Instant Database records in a central location on your network. All data can be shared with all other users without 'activating' the Pathagoras network tools.

So, regardless of whether you have purchased the Pathagoras 'network version,' any single license, stand-alone version of Pathagoras can use any folder anywhere with no setup beyond pointing the program element to the proper location.

The above notwithstanding, if you have purchased a 3-pack (or greater quantity) of Pathagoras licenses, you are automatically eligible for additional networking features of the program. Among these additional features are these:

- a. You can share [profiles](#)⁵³⁴ and [libraries](#)⁵ that you have created with other users.
- b. You can share Instant Database '[Input Form masks](#)²⁶⁰' and '[Matter records](#)¹⁹².'
- c. You can share '[prefix pointer](#)⁴²⁴' assignments. A 'prefix pointer' is a reference to a folder or a glossary associated with a designated prefix. This pointer allows you to instantly recall any clause into your document, regardless of its location, simply by typing its name onto the editing screen and pressing <Alt-G>.
- d. You can share [Multi Choice *List*](#)⁴⁵⁸ assignments.
- e. You can send messages regarding Pathagoras between registered users.
- f. You can [upgrade](#)⁵⁶⁴ Pathagoras via the 'Smart Upgrade' feature.
- g. You can download and keep synchronized with the network source any document assembly text stored in a central location. So, instead of relying on networking connections each time you assemble a document, you can use the always up-to-date local copy of the source text to complete the task. (This feature is perfect for organizations whose staff is frequently in the field and away from direct access to the network files.) See [synchronize text](#)⁵⁶⁵ further down in this section.

So by way of overkill, we want to dissuade you from the notion that the 'network version' of Pathagoras is needed in order to take advantage of network resources. The tools that Pathagoras offers when you have purchased 3 or more licenses are powerful in their own right, but they are for reasons that don't involve accessibility to network resources.

28.1 The Common Profiles Path

Networking and sharing operations discussed in the preceding sections are run through a simple folder called a Common Profiles Path ("CPP"). The CPP holds libraries and profiles, mail and notices, and other common elements and administrative tools that you and other users can store there.

Like other features of Pathagoras, the Common Profiles Path is not an inaccessible series of encoded or encrypted files, forms and folders. Rather, it is a standard, readily accessible, Windows folder. All files in the folder contain plain text references to the network resources.

The most important thing to know about the 'Common Profiles Path' is that there can be only one of them. To function properly, each computer on the Pathagoras network must be pointed to the same CPP.

It is important to understand at this point that while libraries contain 'books,' these books are merely *pointers* to locations which contain the actual source text. The books are not the actual text found at those location. So when we reference the sharing of libraries and books, we speak of the sharing of *pointers* to the various folders or glossaries reflected in the library. When you 'share' a Library, you are sharing the simple 10 line text file of 'pointers' of which the library is composed. When you delete a Library, you merely are deleting the collection of pointers, not any substantive text.

So please recognize that there is never any need to move any documents or other text to your Common Profiles Path. Where it currently resides is just fine.

28.2 Enabling the Network

Part 1: Install Pathagoras.

If you have not already done so, install Pathagoras on each computer you intend to make part of the network.

Part 2: Activate the Network

1. In the 'Utilities/Settings' screen click the 'Networking Setup' tab. (Figure 1, below).

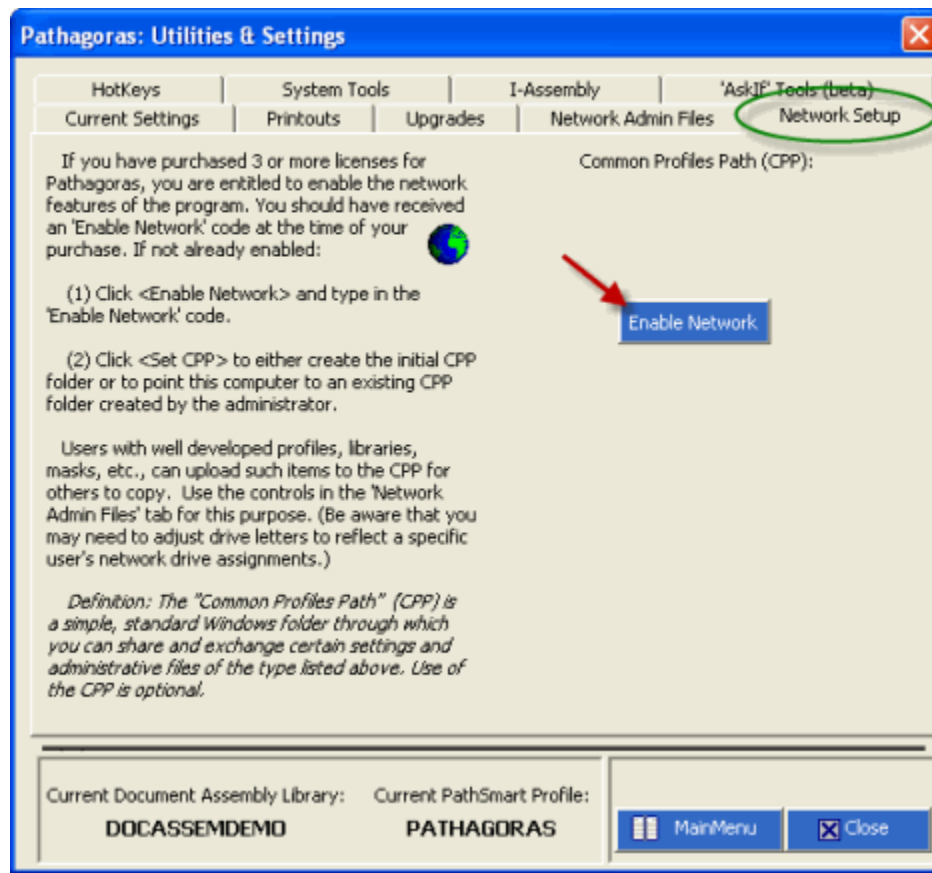


Figure 1. Network Setup

2. Click <Enable Network> (center of page toward the right).
3. When prompted, type into the popup screen the unlock code that you have been provided. Your unlock code was sent to you by separate correspondence. If you do not have it, write us and we will generate another code and send it to you.

Part 2: Establish a Common Profiles Path

Once you have enabled the network with the proper password, a new set of objects appear on the screen:

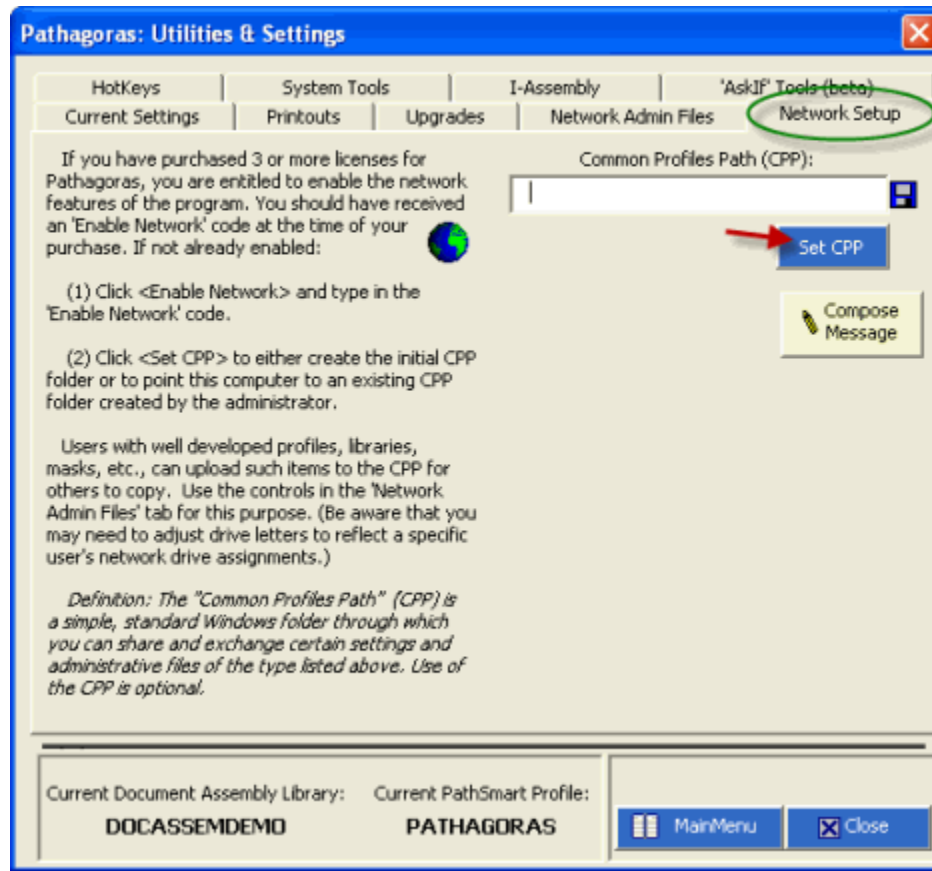


Figure 2. Network Setup Screen (after network has been enabled).

1. If you have already created a location for your Common Profiles Path, you can hand type its location in the white text box. When you are done, click the 'save' disk that appears when you begin typing. Skip to step 4 below. Alternatively, you can click the <Set CPP> button immediately beneath the text box. Follow the instructions in steps 2 & 3 below.
2. If you have not hand-typed the CPP address, click the <Set CPP> button. Following the prompts, navigate to the folder that you want to be your Common Profiles Path (CPP). If you have not previously created a folder to serve as the CPP, you can create one 'on the fly' as you are navigating to your ultimate destination.
3. When you have navigated to the desired folder, click OK.
4. When the CPP has been selected, click the <Register> button that appears just beneath the CPP path. This creates a subfolder within the CPP unique to the specific computer on which you are operating. Messages regarding Pathagoras (e.g., program upgrades -- see below -- or modifications to profiles and libraries) are sent and received through this subfolder.
5. We suggest that you write down on a piece of paper the full path to the CPP folder. This will be for the benefit of those (including yourself) who will need to point to it later when you are setting up the other computers.
6. Click the button that will next appear called "Register." This sends your computers formal name to the Common Profiles Path, and registers you as an authorized user.

Immediately after this step, Pathagoras will ask if you want to download any existing libraries and profiles to your local computer. If you have a 'mature' system, one in which libraries and other settings have been uploaded from a remote computer to the common profiles path, you typically should say "yes." Of course, you can say 'yes' even if this is a new system, but there will be nothing to download. (Saying 'yes' will cause any libraries which may have been uploaded by a user to the common profiles for sharing with others to be downloaded onto the 'current' computer, thus saving the time necessary to recreate that same library. See [Network Files](#) ⁵⁶⁰.)

Repeat the above steps for each computer on the network for which you have purchased a license. Of course, instead of creating a Common Profiles Path for the second and subsequent installations, you will simply navigate to the CPP. Hence, the piece of paper suggested in step 5 above.

Notes on the Common Profiles Path:

The CPP can be anywhere, but there can be only one CPP on the network. Therefore, the CPP should be on your network file server or other location readily accessible to all users.

We suggest creating and using a folder called "PathagorasCPP", but any name is acceptable.

Do not 'overthink' what the CPP is or does. The CPP is nothing more than a folder to which the other computers on your network will refer for information, data, messaging, etc.

While you can store your document assembly books within the Common Profiles Path, it is certainly not necessary that you do so. Documents, glossaries, books, forms, etc. can be stored (and called by the Document Assembly system from) anywhere.

28.3 Network Features

There are a myriad of new features that can be accessed once the network has been enabled. The following screens discuss them.

28.3.1 Sharing Files

Sharing Libraries and Profiles

Uploading

Let's assume that you have created on your local computer the 'perfect' Library for document assembly purposes. You now want to share it with others on the network. (That way, of course, those users won't have to manually create the same setup.) You should upload your masterpieces to the Common Profiles Path (the CPP). Here are the steps:

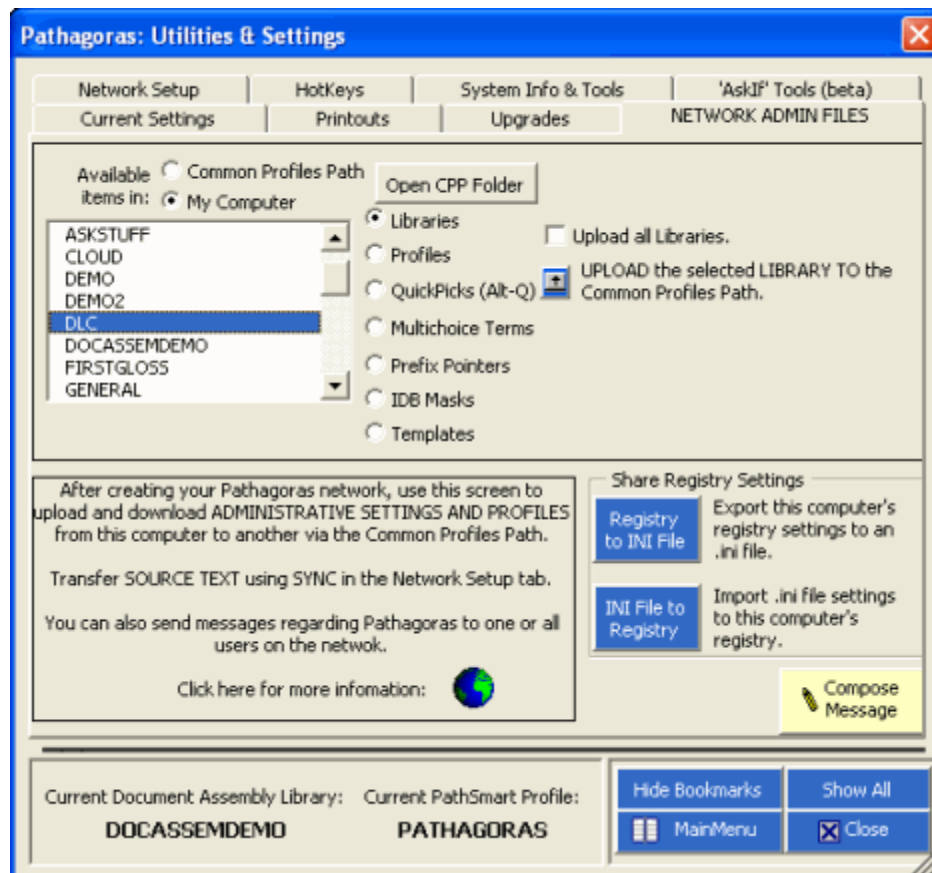


Figure 1. Network Files Screen

1. Click Utilities/Settings>>Network Files to display the Network Files screen (Figure 1). Choose whether you want to view the items in the Common Profiles path (which you would want to do if you want to download someone else's work from the CPP) or to view items in 'My Computer' (which you would want to do if you want to upload your work to the CPP). Note the elements on the screen that automatically change based on your selection to reflect the above discussion.
2. Select the option circle reflecting the element you want to upload to the CPP.
3. Press the up arrow (UPLOAD) to transfer a copy of your work up to the CPP.



Note: Whenever you modify a Library using the Document Assembly "Settings" screen or a Profile in the PathSmart "Settings" screen, a black and yellow button labeled "Send Notice" will appear at the right side of the settings screen after you press the <Save> button. If you click <Send Notice>, a copy of the new/modified Library will be sent to the CPP and a notice of the change will be sent to all registered users. This accomplishes a bit more directly what steps 1-3 above accomplish.

Downloading:

1. Other users on the system can now get your 'stuff' in almost the same fashion.
2. Display the "Network Tools" screen and select the Common Profiles Path option and then select the type of item you wish to download.
3. The list changes depending upon the selections made. Click on the item from the list

that you want to transfer and then click the Download button. (See Figure 3 below.)

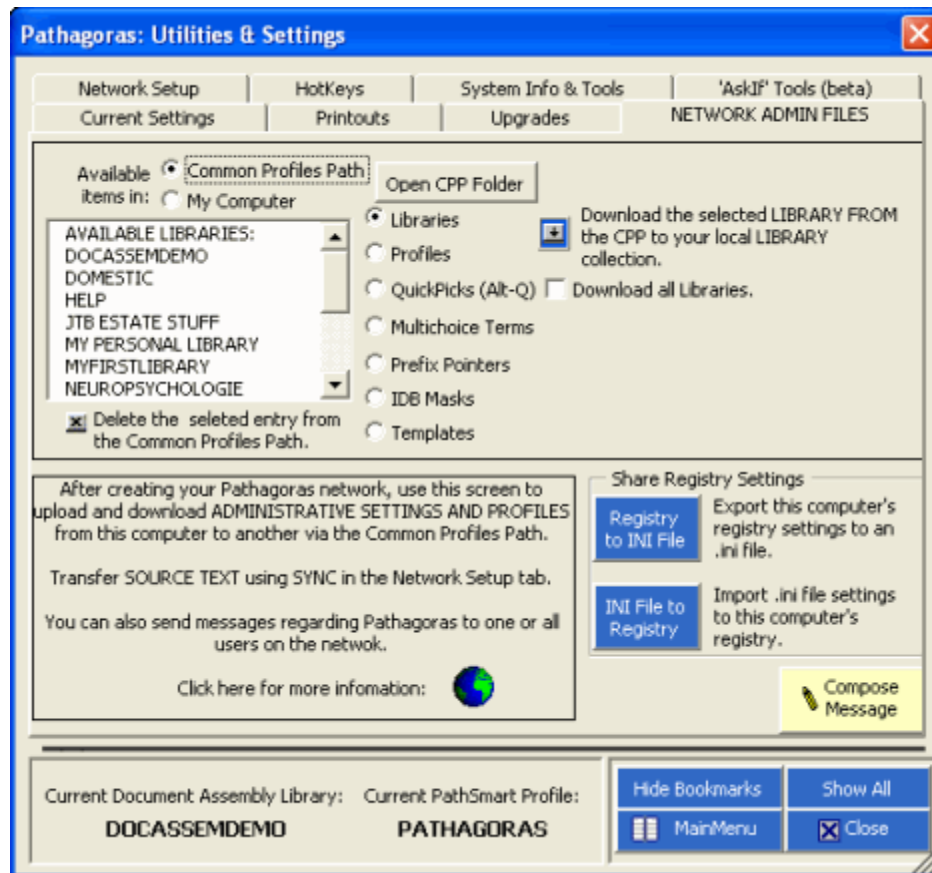


Figure 3. Network Tools, downloading a profile.

➔ **IMPORTANT:** Be mindful of the differences between 'universal' and 'specific' folder references:

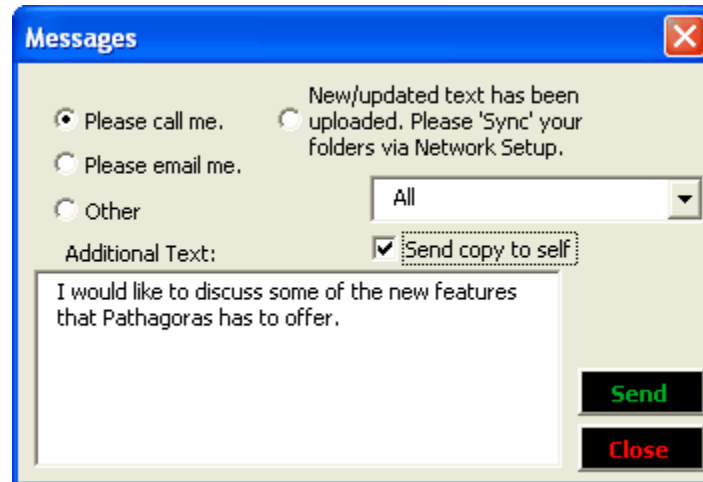
- If the uploaded profile or library refers to the originator's local computer, e.g., "C:\office forms\contracts," a user on a distant computer downloading this profile will not be able to use the file without first adjusting the references.
- This can be done after the download using Document Assembly>>Settings (if a library) or PathSmart>>Settings (if a profile). Edit the entries to reflect the correct network path to the originator's computer, e.g., "G:\office forms\contracts" or "\\server1\office forms\contracts".
- This problem can be avoided on the 'upload side.' Before sending the profile or library to the CPP, the operator should convert local 'C:\' references to 'universal' path references. E.g., "G:\office forms\contracts" or "\\server1\office forms\contracts".

Sharing other Settings:

As the images above suggest, you can upload and download a wide range of settings and data elements via the CPP, including Libraries, Profiles, DropDown Lists, MultiChoice Term lists, Prefix Pointer tables, Q & A Interviews, IDB Masks and Templates.

28.3.2 Messaging

You can communicate with other Pathagoras users by clicking the Send Message button on either of the Network screens. When you do so, you will see a screen that looks like this:



You can select a recipient -one, several or as shown above, 'all' registered users. (You can also choose to send a copy to yourself.)

You can select one of the pre-composed messages and/or compose your own message.

The next time a user tries to access a Pathagoras feature, a screen will appear on the user's computer announcing that 'You have a message from Xxxxx.' The user can accept the message or come back to it when more convenient.

28.4 Instant Database Issues

The question frequently arises, after a Pathagoras network is established:


"Ok, now what do I do with my Instant Database files?"

And the answer is, "Anything you want."

- You can leave all IDB records where they are.
- You can move all IDB records to a common location.
- You can share your local records, or not.
- You can share the matter records and not the masks, and *vice versa*.
- You can maintain one set of matter records or masks, or both, on your local drive and another set on the network. That way, you can keep some records and masks 'personal' and others you can share on the network.

Pathagoras imposes no requirements, just alternatives. (We know that Pathagoras' steadfast refusal to impose 'this is the way it has to be done' requirements is unsatisfactory to many. But hey, that's what makes us so popular.)

So, you have to decide what you want to do. Once you make that decision, Pathagoras


makes it easy to implement. You can set and change Pathagoras file locations, and decide what you are going to keep local and what you are going to keep in a common location, from the [Instant Database File Locations](#)  screen. Click the link to be taken to the page that discusses it.

The Network files tab under the Utilities/Settings screen allows you to upload and download Masks and Matter records to and from the Common Profiles Path. Be mindful of the difference between uploading an InstantDatabase mask to the Common Profiles Path (or downloading one from the CPP) and 'sharing a common Instant Database Mask'.

- a. In the former situation, you are simply transferring the 'physical' record from one location to another. A file that was downloaded from the CPP will look like the original file that was uploaded, but it is not 'the same file.' You are not technically *sharing* the file, so much as you are working from a file that looks like the original. You can make changes to the original or copy of the Mask or Matter record, and it will not affect the other copy. (And this would explain why, when one user changes the mask and the other user who had a copy of the mask, does not see the change.)
- b. In the 'sharing' scenario, the IDB pointers on the individual computers all point to the same file on the network. The end users (and the administrator) should all set their IDB file pointer to a common location. This is done via the InstantDatabase Tools|File Locations page. A change to the mask or the matter record will be seen by all users who share the common mask.
- c. You might prefer to transfer matters and masks to and from the CPP if (1) if most users on the network operate as stand alone operations, rarely sharing data; or (2) are frequently away from the network or (3) rarely have a need to use the masks created by another user. In these situations, the administrator should upload 'community' masks to the CPP. The end-users should be instructed to download one or more of the masks to their local drives. When changes are made to a mask, the revised masks are sent up to the CPP, overwriting the original copy, and end-users are advised of the change and instructed to download the newest version.

28.5 Upgrading Pathagoras

Periodically, Pathagoras is updated to provide new features, enhancement to existing features, or just plain bug fixes. Multi-licensed users can take advantage of the Smart Upgrade feature to avoid having to download more than one copy of the upgrade package from the Internet. Here are the steps to perform when you receive an upgrade notice.

➔ These instructions assume that all users on the network have properly installed Pathagoras and have activated the network functions. It also assumes that all users are pointing to the same Common Profiles Path. If you have not done so, please visit [Enabling the Network](#) .

1. One user on the network should be designated the 'initial upgrader.' That user will be referred to as 'you' in the next steps.
2. From the Utilities/Settings>>Upgrade screen, click the "Download Upgrade Package from Internet" button.
3. Pathagoras will connect to www.pathagoras.com and will begin downloading the upgrade.

4. When prompted by your browser, choose to "SAVE" the upgrade file (it will be called "Pathagoras Setup.exe"), and then save it to your Desktop. The desktop is typically the default 'save to' location for most browsers.
 - ➔ It is important that you save Pathagoras Setup.exe to the DESKTOP in order for the notification steps in paragraph 9 below to be triggered. (Pathagoras provides reminders of this step during the upgrade process.)
5. Close Word and Outlook completely. You can do this while the download completes.
6. When the download is complete, run Pathagoras Setup.exe from the desktop.
7. A reminder for you to fully close Word and Outlook (the step 5 actions) will appear on the screen. When you are ready, press OK to continue the install.
8. The next screens will be identical to those you have seen (license notice, user info, etc.) in previous installations of Pathagoras. When asked if you want to start Word after installation, say 'Yes'.
9. When Word reopens, you will be asked these 3 questions. Unless there is a reason to say 'no,' answer each 'Yes.'
 - "Do you want to transfer the upgrade to the Common Profiles Path?"
 - "Do you want to notify other registered users of the update?"
 - "Remove Setup from Desktop?"
10. The next time a registered user on the network begins a document assembly or document management project, Pathagoras will advise the user that an update is available and ask permission to install it. Unless there is a reason to say "No," answer "Yes."
 - ➔ If the User answers "No," he or she should perform a "Smart Upgrade" as soon as possible. A Smart Upgrade is performed from the Utilities/ Settings>>Upgrade screen. The <Smart Upgrade> button is at the right side of the screen.

28.6 Synchronizing Books

A 'book' is a pointer to a folder or a glossary. These folders and glossaries in turn contain the source clauses used for document assembly purposes. In a typical setup, all of the users on a network point to same source. The clauses contained in that book are physically shared among all the users.

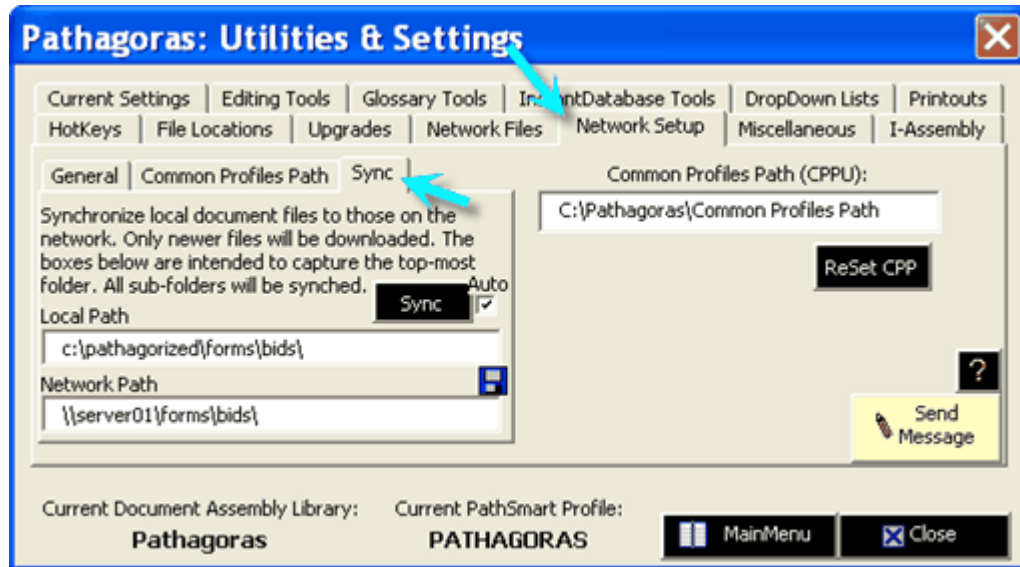
However, some organizations, or certain users within the organization, find that physically sharing source clauses is not practical. Sometimes it is a slow network that leads to the decision. Other times, it is because it is impossible to be physically connected to the server at the time the clauses are needed. Organizations whose staff is frequently in the 'field' and away from direct access to the network files is a good example of the latter.

Pathagoras allows registered network users to synchronize 'centralized' source text to a local copy of the same on the remote computer. So, instead of relying on networking connections each time you assemble a document, you can use the up-to-date local copy of the source text to complete the task.

- a. In the Utilities/Settings menu, click the Sync tab in the Network Setup screen.
- b. Complete the two text boxes representing the local path (which will receive the latest

files posted by the administrator and the Network path where the latest files are maintained. You should type the top-most folder where the two systems 'match up.' All sub-folders beneath will be synched.

It doesn't really matter what either folder is called. It is a bit easier to visualize the process if at least the last named sub-folder of each are identically named.



The Local Path and the Network Path are paired for synchronization.

- ➔ The synch is 'one-way only' -- *from* the network *to* the remote computer. Only one person (an administrator) should be in charge of updating the original material.
- ➔ Only newer files will be downloaded. So the download is quick and only when necessary.
- ➔ Check the 'Auto' box to automatically synchronize files at the start of each Word session.

Some users may wish to use locally stored files even if the network files are available. This can dramatically speed up processing. See next section titled "Use Local" for more information on this useful feature. (Of course, you must be absolutely confident that the local files accurately reflect the contents of the network files.)

28.7 'Use Local' Files

Pathagoras easily connects to network servers, and even in its stand-alone version, can use any and all files stored on the network server.

However, you are not always connected to the network server. So it would be 'nice,' would it not, if you could 'default' to using local copies of the files downloaded and 'synched' to the network?

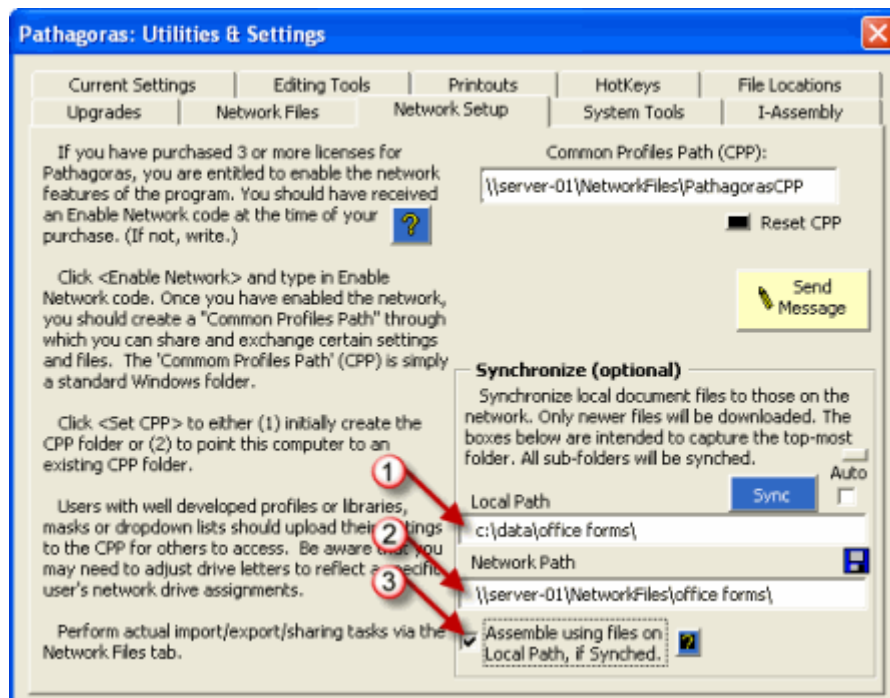
Further, even if you are always connected to the server, document assembly proceeds at a much faster pace when files are stored on and are assembled from the local computer. Wouldn't it be nice if you could 'default' to using the local file copies of the network files?

Pathagoras offers this in its "Use Local" feature. It is one of those 'best of both worlds' features that Pathagoras offers where possible. 'Use Local' allows you to set up your libraries & book as if you are running completely off the network, yet, if the files are also stored locally, the actually call on source text is made to those local file. So, point all of your books in your

Libraries and Settings screen to the network server (that way it is easier to 'share' libraries since they will be pointing to 'universally recognized' locations) and populate all of your DropDown Lists with files compiled from the server. But turn on the "Use Local" switch (after making sure that you have 'Synched' your local data to the Network data.

If you want to use this feature, you must have purchased at least the 3-pack of Pathagoras, and enabled the Pathagoras Network features. This is found in the Network Settings tab in the Utilities/Settings screen, shown below.

Then, on the same screen, point Pathagoras to both the Local (1) and Network (2) file folders. Lastly, check the switch "Assemble using files on Local Path, if Synched" (3) to create the default assembly action.



Synchronizing (previous section) and "Use Local" are similar but not identical concepts. When you synchronize folders/books, you are copying clauses, forms and templates from the network server to the local computer. This is identical. But with 'plain' synchronizing, the books always point to the local hard drive. With "Use Local," the books point to the network drive, but when a document assembly call is made, Pathagoras checks to see if the identically named file resides on the local drive. If it does, it is used instead of the copy on the network. This allows for continuous connection to the network, but faster speed in assembly by using whatever local files may exist. It also points up the need for synchronized books. If "Use Local" is on, but the local document is not the latest version, the assembled document will be 'out-of-date' (and perhaps dead wrong).

The Pathagoras System

Settings

Part



XXIX

29 Settings

You will find Pathagoras to be a broad, functional and flexible program. However, in order to allow the flexibility it offers, Pathagoras provides dozens of possible settings and switches which can be changed at the desire of the end user.

Any time that choices are presented, anxiety occurs. "Is this the best setting?" "What does this setting really mean?" "Am I doing this right?"

Rest assured, the default settings are more than adequate and appropriate for the beginner. We recommend making no changes to any of the default setting. When you begin asking the question "I wonder if Pathagoras can do this?", then begin exploring the options.

What are some settings?

- **Instant Database**

- Can Pathagoras automatically scan assembled documents for variables without a user request? Yes
- If I use variables with long names, can I change the width of the Instant Database screen to accommodate them?
- If I use lots of variables, can I change the height of the Instant Database screen to show more of them with scrolling?
- Can I move my IDB records to my server so that everyone in my office can use them?

- **Document Assembly:**

- If I use the prefix/suffix naming convention, can I show the names or subjects by default.

- **DropDown Lists**

- Can I set the List to show the names instead of the subjects, or the subjects instead of the names?

Other settings involve pointing a book to a folder. These are more 'substantive' settings because they control what Pathagoras will display (as opposed to how it displays it).

- **Document Assembly:**

- How do I create a new library?
- How do I add a book?

Settings can be accessed in a variety of fashions. Regardless of the access point, a change made to a setting from one of the available screens will be disseminated to all sections of the program. That's because the settings are actually stored in your computer's registry, not in the individual screens from which your settings may be modified.

Main Menu

As is appropriate for a Main Menu, you can access and select many of Pathagoras settings. The settings visible here are displayed in what we intend to be an 'educational' fashion. We really don't expect you to make any changes via this screen (although you certainly may), but rather we just want to expose you to some important settings and features of the program that you should know about and may wish to invoke as you become a more experienced user. Each setting contains a brief explanation of the purpose of the setting and the 'result' associated with

selecting one of the alternative values.

Functional screens

As you invoke a program module (Document Assembly, Instant Database, PathSmart, etc.), a screen will appear to take you through the desired operation. The screen, or screens that can be invoked from that screen, will contain or lead you to settings that are integrally related to that module. Make settings as are appropriate.

Utilities/Settings Screen

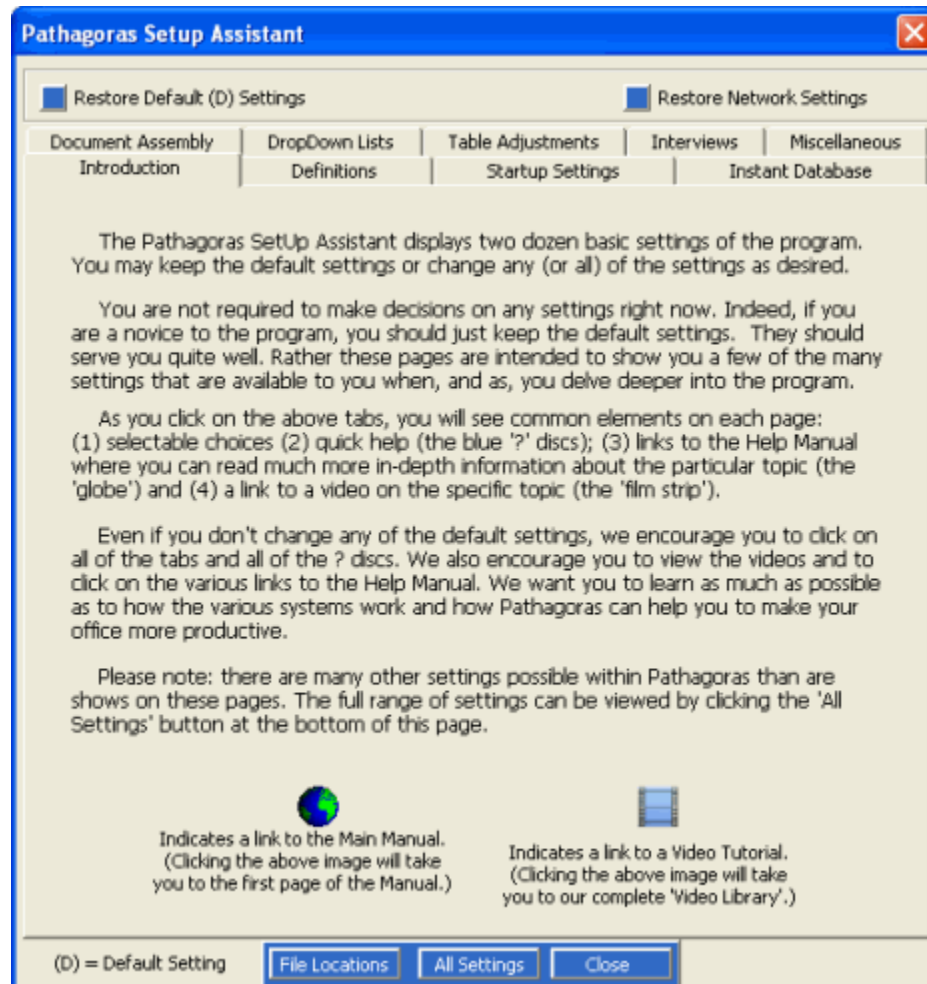
The "Utilities" side of this screen allow you to run certain administrative tasks that cannot be conveniently placed elsewhere, but many 'settings' that have been made (or that are in place by default) are also displayed. You can see these in the section titled "*Settings at a Glance.*"

All Settings Screen

this is the most comprehensive listing of the settings available in the program. By its nature, it is the most complex as well.

29.1 Settings Assistant

This screen, accessed from the Main Menu, is a brief summary of a couple of dozen of the primary settings used in Pathagoras operations. The screen itself describes its use and purpose.



29.2 Utilities & Settings

This is the 'guts' of the program. If it isn't somewhere else, it's here, or accessible from here.

Most Utilities/Settings topics (reflected by the tabs on the screen) are discussed in separate sections elsewhere in this Manual. headings. See:

[Editing Tools](#)  446

[Instant Database Tools](#)  212

[DropDown Lists](#)  356

[hot-keys](#)  590

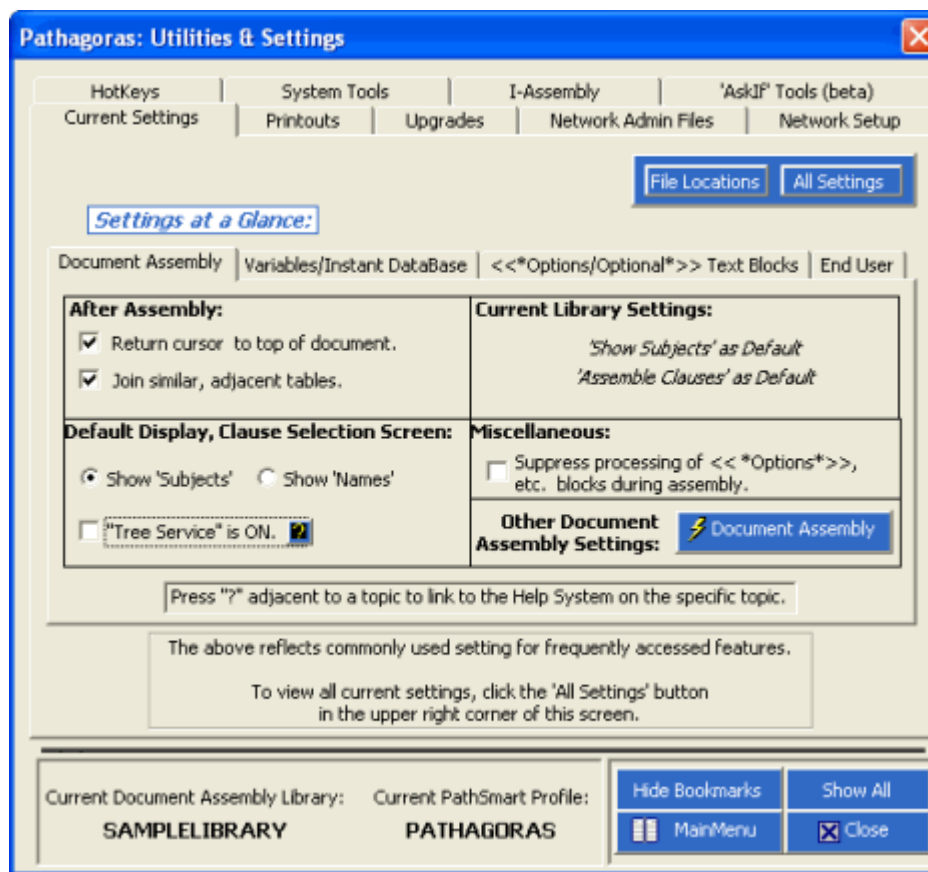
[Upgrading Pathagoras](#)  564

[Network features and functions](#)  556

29.2.1 Current Settings

The initial page of the Utilities/Settings screen. Note the following aspects.

- The 'Current Settings' tab contains a "Settings at a Glance" section, which allows you to quickly view and modify the most frequently accessed program settings.
- Clicking the <All Settings> button in the upper right quadrant of the screen will take you to another screen. There, you can view and modify all program settings. This is discussed in a separate section of this manual.
- If you want to see where the variety of administrative pointers 'point', click the File Locations button.



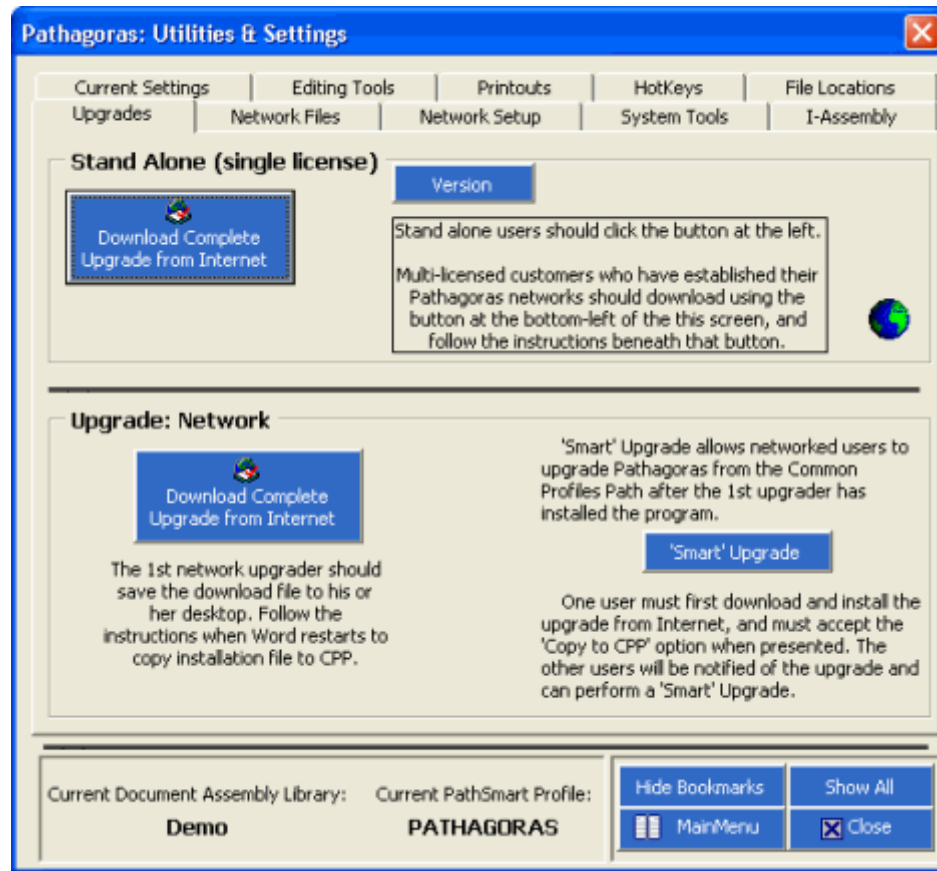
Utilities/Settings: Current Settings

- The bottom left quadrant of the screen displays the current Document Assembly Library and the currently selected PathSmart Profile.
- The bottom right quadrant of the screen provides a series of action buttons with self-explanatory titles.

29.2.2 Upgrades

Periodically, Pathagoras is updated to provide new features, enhancements to existing features, or just plain bug fixes. Notices of such upgrades are sent out to all users when it is released. Select the upgrade tab and the appropriate button to start the upgrade:

- Single license and stand alone users should click the 'Download' button in the top section to download the latest version.
- The lower section is designed for those who have purchased multiple licenses of the program and have set a Common Profiles Path for the networked users. The steps to setup the network and perform upgrades are noted on the screen and under the [Networking features](#) ⁵⁶⁰ section of this manual.



Utilities/Settings: Upgrades

Upgrading Pathagoras is simple:

- Pathagoras preserves all libraries, books, settings, Instant Database records, etc. You will lose nothing when you upgrade.
- You do not have to uninstall a previous version of Pathagoras to upgrade.
- The transition from a demo version to a licensed version is seamless.
- The transition from Word2000-2003 to Word2007/2010 is also seamless, although some users will have to reinstall Pathagoras to allow Word2007/2010 to reconfigure itself. But, as stated above, you will lose nothing.

29.2.3 System Tools

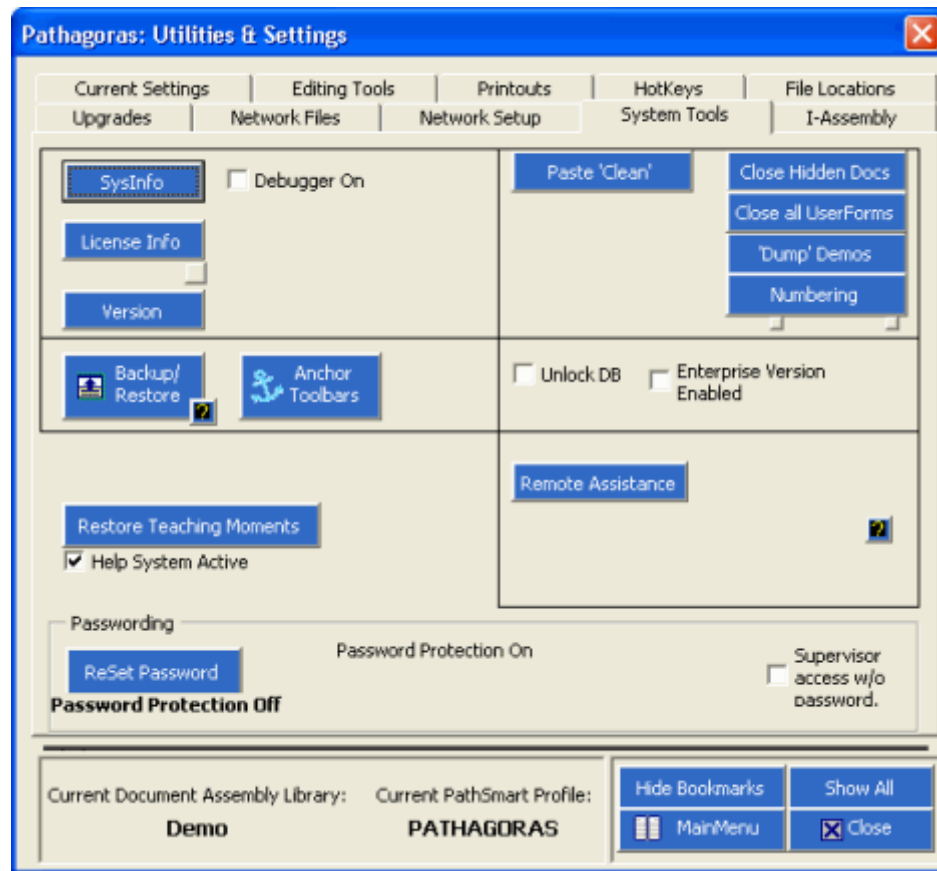
The 'System Tools' tab contains several miscellaneous tools and utilities which cannot be otherwise categorized.

If you wish to backup your settings, perhaps to transfer them to another computer, do so via this screen

If you report a problem with Pathagoras, we may ask you to visit this screen to generate a report that we can use to further define the situation, or to establish a remote assistance session.

Some users have told us that after their learning process has been completed, they want to rid the program of the Demo files that ship with the trial and retail versions. That can be done here,

too. Click "Dump Demos."



Miscellaneous System Tools

29.3 End User Settings

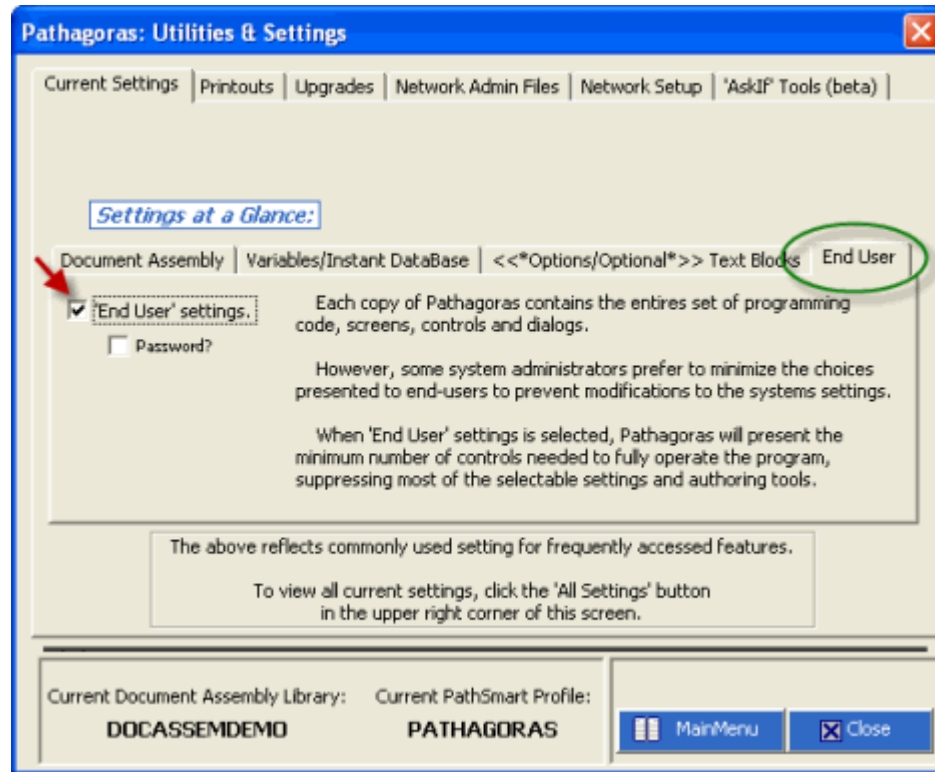
Sometimes there are just too many choices in a program. Pathagoras is not an exception.

As an author/editor/author, when you are preparing your documents for future assembly, you likely want all options, check boxes and authoring tools available. But does not mean that the end users (the ones who are charged with the actual document assembly for the client or customer) need the same information.

We have addressed the 'choice overload' issue by implementing a 'End User' setting that reduces the number of visible options to a bare minimum. It is our intention that, without the 'distractions' of the authoring tools needed by the administrator, it should make explaining Pathagoras to the pure 'end user' a bit easier.

To activate the "End User" switch, click the Pathagoras dropdown features menu. Click the Utilities/Settings button.

On the first screen that you see, look for the new checkbox that says: "Apply End User" (see image below). Check it.



Result: When the End User switch is set, the operator sitting at that computer will not be able to view or activate most of the authoring, clause editing and administrative settings tools. There are two goals of the End User settings:

1. to prove the end user with access to Pathagoras' operational essentials, and deny access to aspects of the program that the administrator wants to declare 'off-limits'.
2. to minimize end user 'overload' of choices that can (and should) be made by the administrator and document authors. (This 'overload' is a good reason in an of itself to enable the 'End User' setting.)

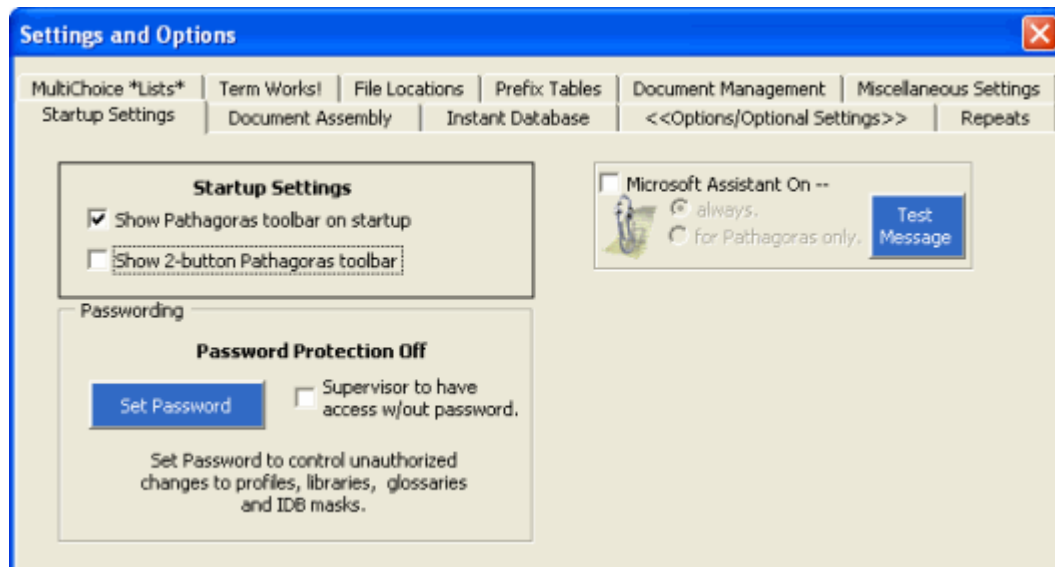
To restore full functionality, simply uncheck the End User checkbox. If you wish to prevent end users from being able to switch to 'on' the various administrative and authoring controls, you should enable password security. Check the box just beneath the End User checkbox. A password will be requested from you as administrator which will required in later sessions to the restore Pathagoras to non-end user status.

See also: [End User Setting as Security Option](#) ⁶⁶⁰

29.4 All Settings and Options

The All Settings screen contains checkboxes and text boxes which allow you to control every display or performance aspect of the program.

Access the All Settings screen via Utilities/Settings. Click the All Settings button. This is the resulting screen.

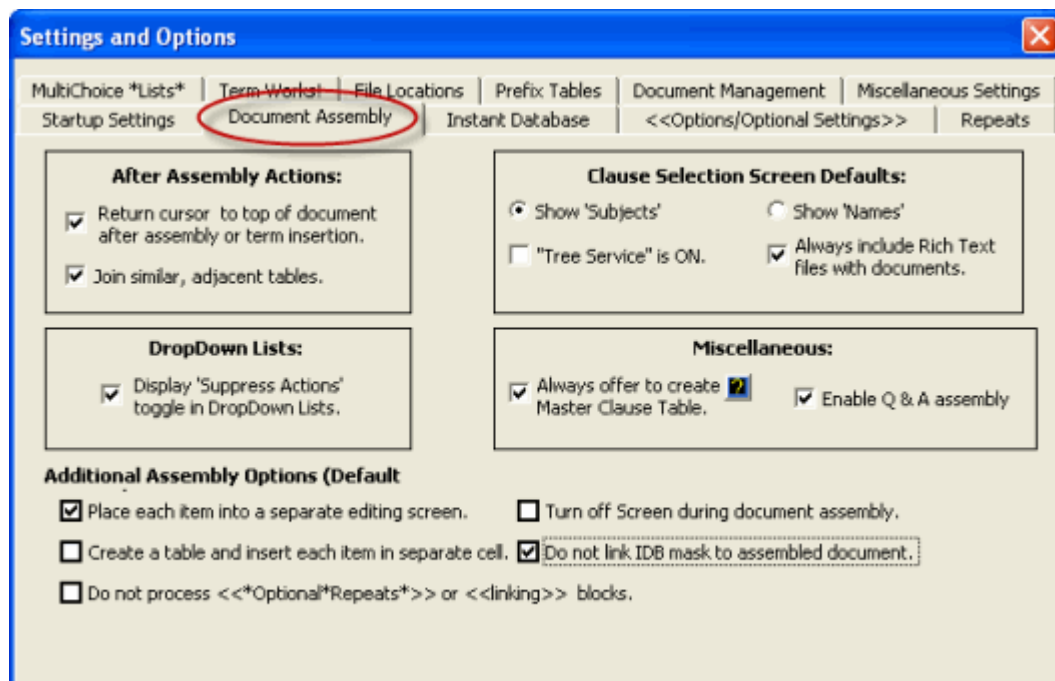


The 'All Settings and Options' Screen. You can check, complete, select, view, modify, etc., every available program setting from this single screen. By perusing the various tabs, you can also explore the numerous settings available so that you can make Pathagoras highly personal to your needs.



You should visit the 'All Settings' screen not only to make changes to program settings you know about, but to explore and learn about those you don't. There likely are many setting and option that you may not have even known existed.

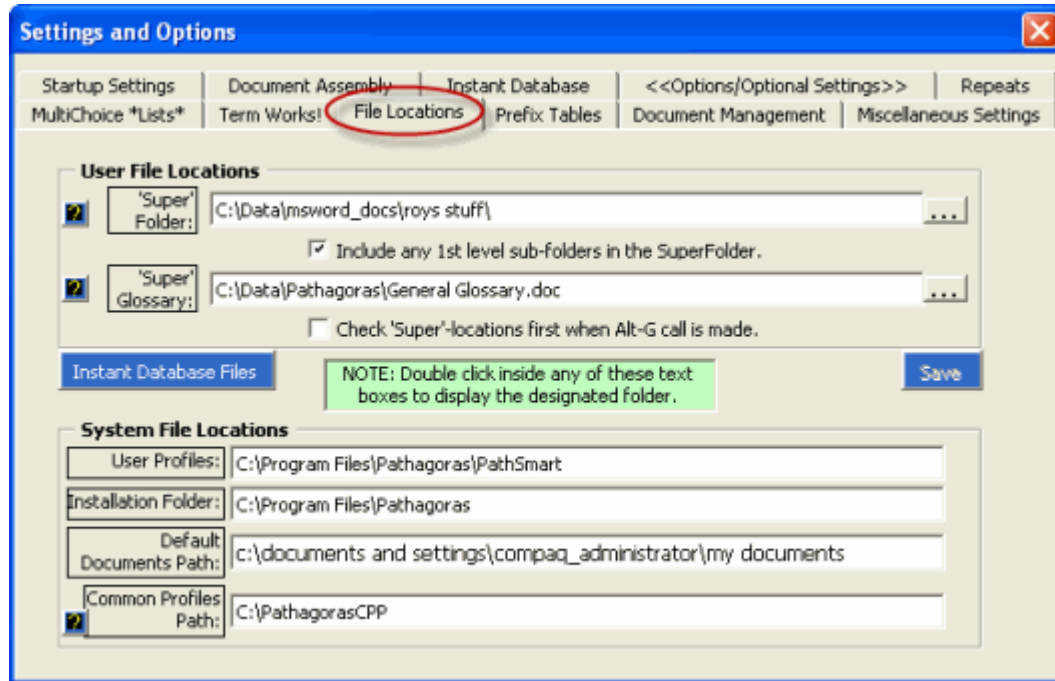
29.4.1 Document Assembly



29.4.2 File Locations

Click the File Locations tab to see where the important program files, folders and elements are located.

A double click one any text box containing a fully qualifies DOS path will take you to that specific path. We don't recommend that you change anything, but it is nice to be able to explore a bit.



Utilities/Settings: File Locations

Note the top two items on the page: Super Folder and Super Glossary. These are [discussed in greater detail here](#)⁹⁰, but both are wonderful tools which allow you to instantly call a piece of text stored in either "Super Book" into the document you are currently editing. The technique? Once either or both SuperBooks are set, you simply need to type onto the editing screen the name of a document located in either SuperBook. Press the hot-key <Alt-G> and Pathagoras will instant retrieve the document and place it onto the editing screen.

Click this link for a lengthier [discussion of file location issues](#)⁶⁵⁷.

29.4.3 Separators and Connectors

When you have a list of items that you want to insert into your document, you may also wish to include an appropriate separator and connector between each element of the list. That way, instead of the results appearing like this:

"red blue green orange"

you can have them automatically read like this

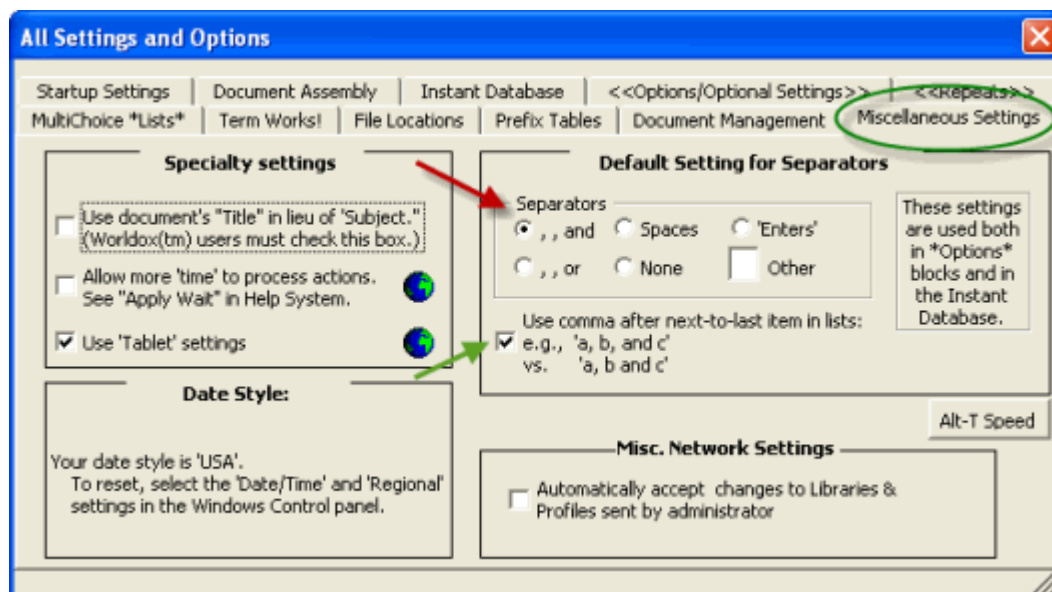
"red, blue, green and orange" or

"red, blue, green or orange" or

"red + blue + green + orange".

Pathagoras can add several different separators to your lists. You can select the separator from the activity screen on which your list appears.

You can set the **default** separators/connectors (the ones that will be selected when the list first displays on an activity screen) from the Miscellaneous tab of the All Settings screen:



29.5 Sharing Registry Settings

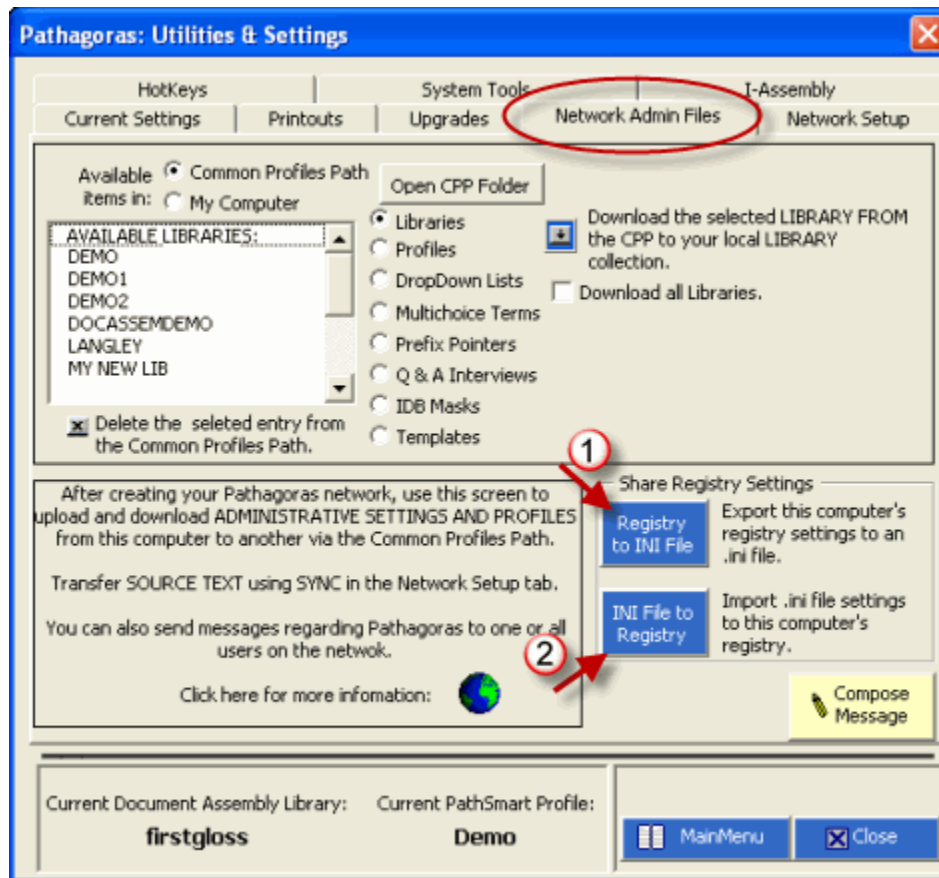
Like in most programs, most default and user-selected settings and pointers that control the operation of Pathagoras are stored in the Windows registry. (There are about 120 user-selected settings available in the program.)

Network administrators who are controlling the installation of multiple computers will find it helpful to establish a Common Profiles Path which can be used to share "Libraries" and "Profiles" and other items not stored within the registry. Sharing such items through the Common Profiles Path is [fully discussed at this link](#).⁵⁵⁷

Network administrators will also want to be able to share the various registry settings that the administrators want to be common to all users. Its operation is similar to that of how the Common Profiles Path operates. A set of values is uploaded by the administrator to a location accessible to the remote users. The remote users in turn download the values. The sharing of registry settings can be used in conjunction with the Common Profiles Path or run without the existence of the Common Profiles Path.

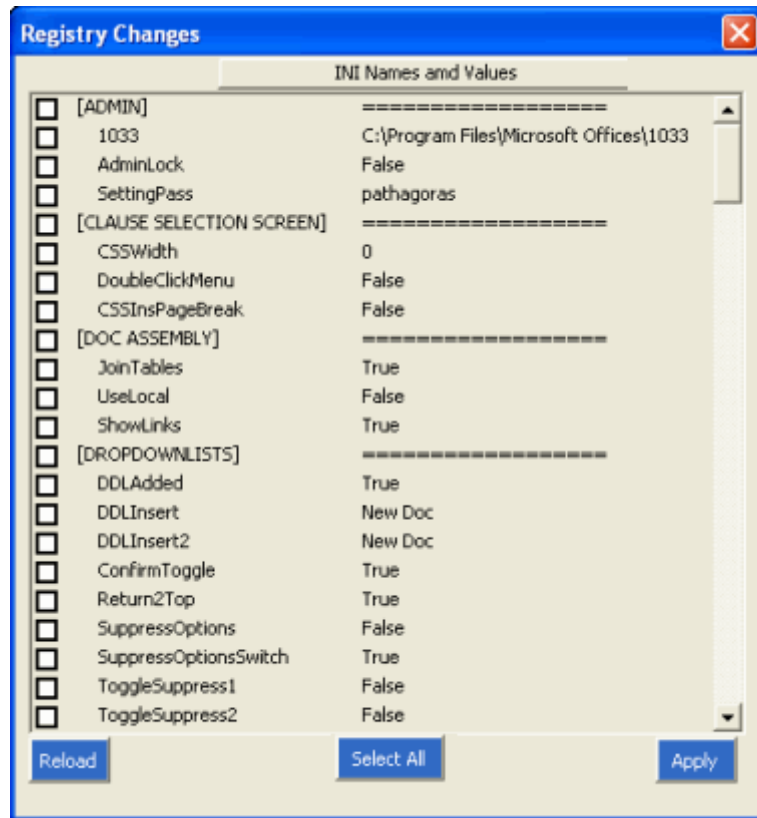
The settings that can be shared are summarized on the [next page](#).⁵⁸¹

The buttons to accomplish the upload and download are found among the other tools on the Network Administrative Files page of Pathagoras Utilities & Settings screen.



Assuming that the administrator's computer currently has the 'preferred settings' for the office, the administrator would click the button labeled "Registry to INI file" (#1 above). When pressed, the all of Pathagoras' registry settings of the administrator's computer are exported to a file called 'pathagoras.ini'. (".ini" is the classic extension assigned to files intended 'initialize' a program at installation time.) You will be given the option to save pathagoras.ini to the Common Profiles path or to any other folder, including a thumb drive, that both computers can access.

Once uploaded to its 'central' location, each remote user would click the lower button on the Utilities/Settings screen labeled "INI File to Registry" (#2 above) to download and then to display the various available entries from the .ini file onto a categorized list of all available settings. The registry 'name' is shown at the left, and the registry value as exported by the administrator in step 1 are shown at the right. The remote user can then choose all or a select few of the available registry settings. The local registry is updated when the <Apply> button is clicked.



With the content of the .ini file displayed, the remote user can select one, all or an appropriate selection of the various registry settings that are available.

Click Apply and registry is updated.

29.6 Registry Settings


The following is an explanation (with links to fuller discussions) of many of the various registry settings that are maintained in the Registry entry called "Pathagoras". Modifying the registry should be done with care and only by system administrators familiar with registry function and structure.

Note: this is not a comprehensive listing of all registry entries pertaining to Pathagoras. These primarily reflect those entries that, in an network environment, might be common among all computers on the Pathagoras network and which could be share among all members without affecting essential local settings.

[ADMIN]

AdminLock	Indicates whether the Administrator on the computer must also provide a password in order to effect settings changes.
SettingPass	The password required to effect settings changes. See System Security 660

[CLAUSE SELECTION SCREEN]

CSSWidth	The default width of the Clause Selection Screen. See Widen Screen ⁵⁹ .
DoubleClickMenu	Indicates whether the Double Click Menu ⁶³ displays by default at the top of the Clause Selection Screen.
IncludeRTF	Include Richtext files in listings. See Files Included in Displays ⁵³
CSSInsPageBreak	Indicates whether the 'Insert Page Break between Selections' under Additional Assembly Options ⁶⁴ is set by default.
ShowLinks	Include shortcut references to documents that reside in other folders. (A shortcut is indicated by an arrow in the lower left corner of the icon: ) See Files Included in Displays ⁵³
UseLinear	Default assembly setting is "Linear" (but individual books can still be set to 'Stacked'). See Linear vs. Stacked ⁶¹ .

[DOC ASSEMBLY]

JoinTables	Indicates whether identically configured adjacent tables (separated by a single blank line) are joined at the conclusion of a document assembly routine.
UseLocal	When books have been synched, use the local book even if the reference is to the network version.
NeverReadSubj	Never include the document's subjects in listings.
SuppressOptions	Whether <<Options/Optional>> text blocks are to be processed.

[DROPDOWNLISTS]

DDLInsert	The default value of the "Insert/New Doc" toggle. (Word2003 and prior, this the 'top bar'.)
DDLInsert2	In Word2003 and prior, same as above, but references the 'lower bar'.
ConfirmToggle	Require a confirmation before accepting a switch between "Process/Suppress" and "Insert/New Doc" toggles.
Return2Top	Sets default action of document cursor after a term called from DDL has been inserted.
SuppressOptionsSwitch	Whether the user is shown the Suppress Options toggle in the DropDown Lists
ToggleSuppress1	The current value of the "Process/Suppress" toggle. (Word2003 and prior, this the 'top bar'.)
ToggleSuppress2	In Word2003 and prior, same as above, but references the 'lower bar'.

[GLOSSARIES]

AutoShowGlossReview	Automatically display the Glossary Review toolbar after a glossary has been opened for editing.
---------------------	---

[GOTFORMS]

GotFormsMax	The 'maximum' number of consecutive underlines that comprise a fill-in field.
GotFormsMin	The 'minimum' number of consecutive underlines that comprise a fill-in field.

[Instant Database]

AllowOR	Allow user to separate a multiple choice variable with the term "/OR".
AutoRunIDB	Automatically run the Instant Database routine after document has been assembled. Mimics press of <Alt-D>.
AutoScan	Automatically scan underlying document for variables. Mimics click of Scan button.
BeginTxtIDB	The character(s) which designates the beginning of a variable.
EndTxtIDB	The character(s) which designates the end of a variable.
DeleteBlankIDB	Whether a variable to which no value has been assigned is removed from the document or remains for later processing.
DontAskSave	When Next on IDB screen is clicked, do not ask if IDB record should be saved or updated.
GroupAsk	Whether user should be asked to process other member of !Group!. (Automatic Ask is default.)
HiliteRemVars	Automatically highlight any variables remaining within document.
IDBDateFormat	Default Date 'style'.
IDBDDLHeight	The default 'height' of the IDB screen.
IDBDDLWidth	The default 'width' of the IDB screen.
IDBExclude	The character that immediately follows the opening character of a variable that signals Pathagoras to ignore the text as a variable.
IDBFinalComma	Insert a comma after the penultimate element of a three or more items selected from a multiple choice or MultiChoice list.
IDBfromCustProps	Enable feed from data stored in document's 'Custom Properties'. (Default=Unchecked)
IDBHeight	The default 'height' of the IDB screen.
IDBLocalAndNet	Show links to both local and network masks and data records, allowing user to retrieve values from all 4 locations..
IDBMatterTree	Allow IDB records to be stored in sub-folders.
IDBMaxSize	The number of characters beyond which Pathagoras will reject a

	bracketed term as a variable.
IDBPassword	Password needed to create or modify a mask.
IDBPathA	The folder that contains archived IDB records.
IDBPathMasks	The folder that contains IDB masks.
IDBPathMasksL	The local folder that contains IDB masks.
IDBPathMasksN	The folder on the server that contains IDB masks.
IDBPathPIRL	The folder that contains IDB personal data records.
IDBPathPIRN	The folder on the server that contains IDB masks.
IDBSeparate	Whether to maintain Masks in a folder separate from Records
IDBSkipWarn	Automatically reject all bracketed term in excess of IDBMaxSize. (Default=Warn before rejecting.)
IDBSlashOr	Allow user to separate a multiple choice variable with the term "/OR". (Use when multiple choice variable contains a slash as part of its 'name'. E.g., a list of dates in m/d/y format.)
IDBTreeDelete	If IDB Tree in use, delete record from original location upon move of record.
IDBWidth	The default 'height' of the IDB screen.
MatterData	The text used to label the personal data list in upper right corner of IDB screen.
MulSelOff	Enables/disables the display of the multiple choice selector
PreserveVar	Whether to preserve variables within document.
ShowIDBCalendar	Allow display of red 'link to calendar' button when 'Date' is part of variable.
ScanNonMainStory	Default setting to scan all aspects of document from variables (including headers, footers, text boxes, etc.)
ProcessFirst	Process any IDB variables within document when 'Process' button in toolbar is pressed.
ProcessFirstAsk	Whether to ask if ProcessFirst to be observed. (Default is to not ask.)
UseEnglishDate	Even if other date format is selected, always use English months for dates

[LISTS]

ExcludeRepeatSeparator

ListSeparator

[MISC]

AssistantOn	Allows Assistant (dog, wizard, etc) to communicate messages. (Word2003 and before)
-------------	--

AutoCreateMCT	Automatically create and update Master Clause Tables
AutoDDLRefresh	Automatically refresh DropDown Lists when new terms added to glossary or folder.
AutoExcel	Automatically export data to an Excel spreadsheet.

[MultiChoice *Lists*]

MCLPath	The folder containing the MultiChoice.cvs file. See MultiChoice Lists
---------	---

[NETWORK]

CPPU	The folder designated as the Common Profiles Path
------	---

[OPTIONS/OPTIONAL TEXT BLOCKS]

QuickOpts	Enable use of '{' and '}' (curly braces) to designate Optional text. This is in addition to those designated below.
BeginTxtOpt	The characters which designate the beginning of an Options/Optional text block.
EndTxtOpt	The characters which designate the end of an Options/Optional text block.

[PathSmart]

DefaultP	
PSWildCard	The 'extra' drive that PathSmart will enable for automatic document searches. (Default='A:')
ShowFiles	Whether files in selected SmartPath will be shown. (Default='Show')

[REGIONAL]

CurFormat	Currency Format
CurSymbol	Currency Symbol
DateFormat	Date Format
DateStyle	DateStyle

[REPEATS]

RepeatSeparator	The character used preceding the incremental numbers (@)
-----------------	--

ExcludeRepeatSeparator	Whether to omit use of the RepeatSeparator (Client1 vs Client@1)
------------------------	--

[SAVESMART]

AutoNameDoc	When other settings in place, automatically propose name for document upon save.
DraftWatermark	Reference to watermark used to denote 'draft'
StampNameonSave	Indicate whether the name of document should be inserted at bottom of document
SSEnabled	Whether to reassign Word's File Save icon to SaveSmart

[SETUP and SPECIAL SETTINGS]

WhichToolbar	4-element vs. 2 element toolbar
TreeDefault	Default on/off setting for Tree Service
EndUser	Default on/off setting for End Users (limited) settings
UseTitle	
Vendor	Used only if special 'vendor'

[SUPERBOOKS]

SuperGlossary	Name of (including full path) Glossary designated as 'SuperGlossary'
SuperPath	Full path to folder designated as 'SuperFolder'
SuperSubFolders	Indicates whether sub-folders to SuperFolder should be inspected for desired document.
SuperGlossFirst	Inspect SuperGlossary before SuperFolder for desired document.

[SYNC]

AutoSync	Indicates whether synchronization between network and local files will occur.
SynchArchive	
SyncLocal	Full path to 'local' folder to which synchronization will be made.
SyncNetwork	Full path to 'network' folder from which synchronization will be made.

[TermWorks]

TWBreakChars	If breakpoint set as 'characters', the character set that Pathagoras looks
--------------	--

	for.
TWBreakHead	If breakpoint set as 'headers,' the level of header (1 or 2) that constitutes a breakpoint.
TWBreakParas	If breakpoint set at 'paragraphs', the number of paragraphs that constitutes a breakpoint.
TWChars	
TWDefault	Current default setting for 'break points'
TWFullScreen	

[WIZARD]

AutoWizard	
WizardRecord	

ForeignLang

FTPView

Help

Images

NoHelpQandA

QandAEnabled

[Misc]

ShowNum	Used for AutoParaNumbering
ShowToolBar	Whether the Pathagoras toolbar will be visible at startup
SingleClick	In PathSmart module, determines whether single click vs. double click opens document.

The Pathagoras System

HotKeys

Part



30 HotKeys

Pathagoras creates 4 'hot-key' assignments to your system upon installation:

- [<Alt-G>](#)⁹⁴: the most used hot-key. Stands for "G"et, as in get a file, or get a folder.
- [<Alt-D>](#)⁵⁹¹: for Instant 'D'atabase.
- [<Alt-Q>](#)¹⁸²: to display QuickPicks (MyButtons) screen.
- [<Alt-V>](#)⁵⁹¹: for Create Variables Assistant.

In addition to hot-keys, Pathagoras provides a series of HotWords which, when typed to the editing screen and followed immediately by the keypress of <Alt-G> will cause certain specific actions to occur:

link<Alt-G>: Link current document to pre-set external data sources.

unlink<Alt-G>: Unlink current document from external data source.

last<Alt-G>: Display last document called or edited via Pathagoras.

lastfolder<Alt-G>: Display folder of last document called or edited via Pathagoras.

sf<Alt-G>: Display your SuperFolder

prefixable<Alt-G>: Display your collection of prefixes and their assigned folders.

repeat<Alt-G>: Recall one more clause of paragraph called via the <<*Repeat*. . .>> function.

Note: <Alt-S> against highlighted text that pairs up with a [QuickLink](#)⁵⁵⁰ name will call the SaveAs screen opened to the QuickLink target.

Learn more about Pathagoras' links to external datasources [by clicking here](#).

CheatSheet: You can print out a 'cheat sheet' of hot-keys and other commands and 'Pathagorizing' markups. To do so, display the Main Menu (from the Pathagoras Features drop down list) and click the "CheatSheet" tab.

30.1 Alt-G

<Alt-G> =

Against un-highlighted text:

- if pressed against a number, Pathagoras will open the SmartPath assigned to that number in the current profile.
- if pressed against a word, Pathagoras will assume that the word is a clause name. It will attempt to locate that name in the locations and in the order designated in the [Search Order](#)⁹⁵ page.

Against highlighted text:

- if pressed against highlighted text, Pathagoras will assume that you want to add the highlighted text to one of your books. It will invoke the [Term Works!](#)³¹¹ routine.

30.2 Alt-S

Alt-S against highlighted text that pairs up with a QuickLink will call the SaveAs screen opened to the [QuickLink](#)⁵⁵⁰ target.

30.3 Alt-D

<Alt-D>=Displays the [Instant Database](#)¹⁸⁶ screen.

30.4 Alt-V

<Alt-V>=Activates the [Create Variables Assistant](#)¹²².

30.5 Changing a Hotkey Assignment

To **reassign** the function to a different hot-key combination

- Click the Pathagoras dropdown menu.
- Click Utilities/Settings.
- Click the hot-keys tab. Change assignments as needed.

31 Typing Expander

The <Alt-G> function discussed in the previous section offers a very powerful 'recall and insert' tool for those documents which you have made part of your document assembly system. But there is really no limit on what you can <Alt-G> into your documents. The power inherent in an <Alt-G> call makes Pathagoras very much like every other typing expanders on the market (ActiveWords™, FastFox™, etc.) as well as Word's built in AutoCorrect feature.

The main difference between Pathagoras' 'typing expander' function and that of the others is that Pathagoras requires you to affirmatively press the <Alt-G> combination. (With the others, the program is designed to 'read' your last few keystrokes and, if a match is found, to respond to what it 'sees'.)

Here are the benefits of the 'affirmative call' requirement of Pathagoras.

- The 'affirmative call' requirement means that you can use real words for your key words. If you want to use a 'real' word such as "letterhead" to reflect the call to your company's letterhead, use it. You don't have to use a cryptic 'ltrhd" or other abbreviation.
- The other expanders are constantly monitoring what you type. If you type any text that matches a 'programmed' group of letters, the expander takes over and gives you what it thinks you want.

Example: You are typing a letter. You want to abbreviate the month of February, but have assigned "Feb" to 'February' in your 'expand' list. You will get "February" when you type 'Feb' whether you want it or not.

This will not happen with the 'affirmative call' requirement of Pathagoras. If you want 'Feb,' type it and move on. If you want February, type Feb and press <Alt-G>.

- Pathagoras allows you not only to call in *text* with <Alt-G>, but also lets you open a folder by simply typing a number or a name.

For this function to work, you must first have assigned a number to a [SmartPath](#)⁵³⁶, or to a name to a [QuickLink](#)⁵⁵⁰. Once you have done that, you need only type the number or its name to the screen and press <Alt-G>. The target folder that you assigned to the number or to the QuickLink will instantly open. No navigation.

- Pathagoras allows you to call in complete, fully formatted documents with ease. Typing expanders are typically intended to call in short phrases, not complete documents, and were surely not designed with document automation in mind.
- You can have the best of both programs. Pathagoras is fully compatible with any typing expander. You can have any of those programs in place and augment them with <Alt-G>.
- Unlike AutoCorrect (which is typically computer bound), everyone on the network can easily point to the same locations to make an <Alt-G> call.
- There is no limit to the type of information you can have available. Images, tables, highly formatted text. An <Alt-G> call can handle it all.
- See [Order of Search](#)⁹⁵ rules to see the methods and the locations Pathagoras will search when you press <Alt-G>.

32 Search Function

Here we want you to think instant Google™ type searches, but the target is not the Internet but rather a search through files in your local system.

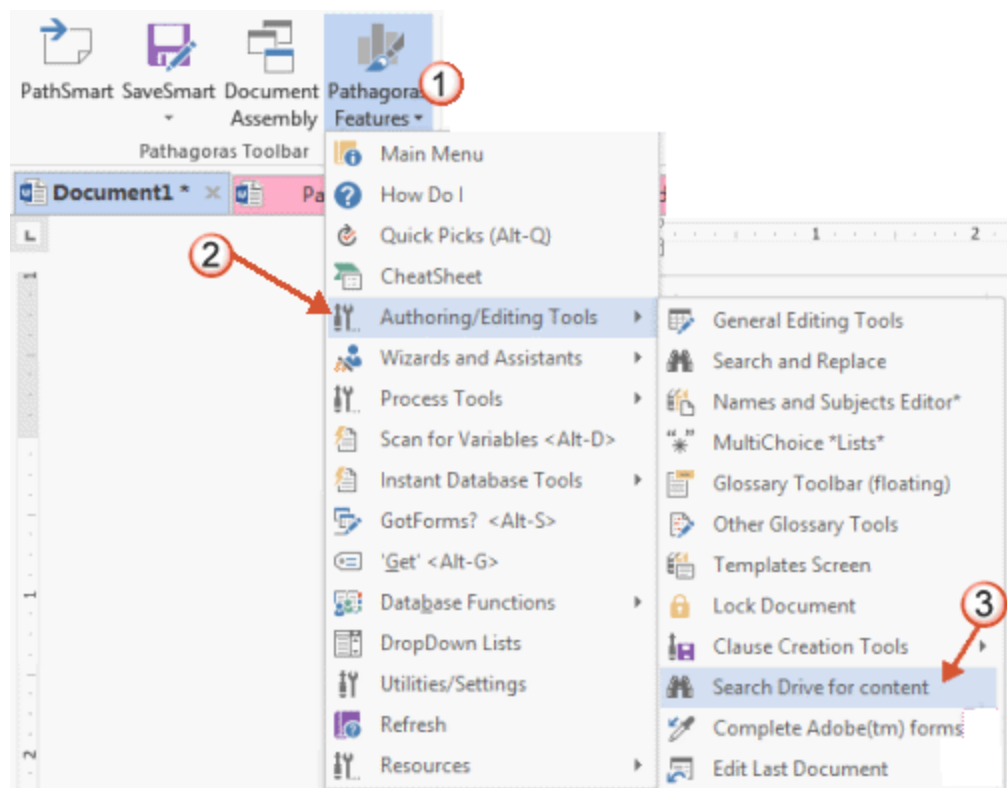
Windows™ offers a very powerful search capabilities. Depending upon your settings, the search can be at the folder level, the file level and (most useful) at the document content level. When properly configured, Windows can perform full-text searches and even index (e.g., Google-style) every word of every file, document, email, etc. on your computer for instant retrieval.

Windows lets you set up filters and complex search criteria. Much is possible with Windows search tools, and you have already paid for the capabilities. So you should take advantage of it. And Pathagoras lets you do just that.

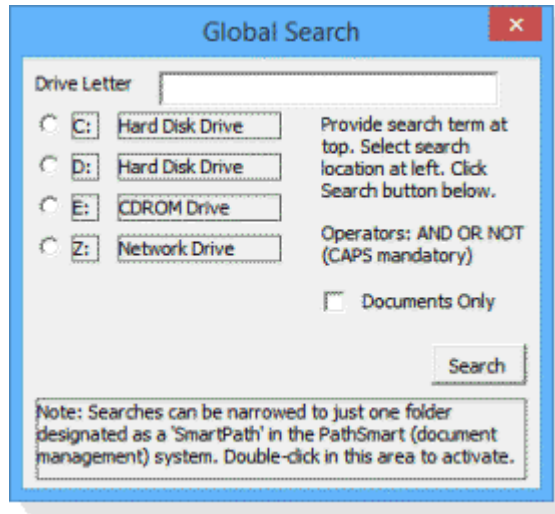
So, when you want to know 'where is that document I wrote 7 years ago that I need now. I haven't the slightest idea where I stored it, but I know it contains the words 'bouillabaisse' and 'refried beans!' Well, Windows can find it if you previously enabled 'full content' indexing.

And now, you can use Pathagoras to do the same. Actually, Pathagoras simply takes advantage of the Windows search function, but does so in a more efficient fashion.

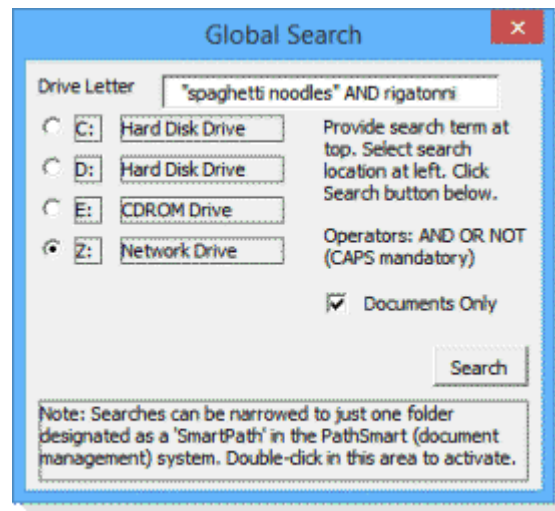
To activate the Search tool, click Pathagoras Features | Authoring/Editing tools. Select the Search Drive for Content entry.



The Global Search screen will appear.



All of your systems drives will appear in the list. Select a Drive to search and provide a search term. Multiple terms can be selected using standard search logic found in other programs. You can restrict your search to just documents by checking the 'Documents Only' box.



Notes: You don't have to exit Word to hunt for your document.

Your search doesn't start until you say 'Search.' (With regular Windows search, a slight pause in your typing in a search term will trigger Windows to 'start looking'. You then are blocked for a second or two, or 10, from completing the search term while Windows hunts for all documents it *thinks* you are looking for. The logic is to prepare for the next letter, but it's frustrating being locked out, especially if you made a typo in the letters.

You can restrict your search to a designated folder by enabling the Search function on the PathSmart screen.



If you anticipate using this feature frequently, you should place the its 'button' onto your Quick Access Toolbar (the 'QAT'). Just right click on the 'Search Drive' button and select the

QAT option. That will provide instant access to this feature, and save you many clicks.

The Pathagoras System

Navigating with Pathagoras

Part




33 Navigating with Pathagoras

Pathagoras essential operations can be summarized as a simple 'pointer' system. A 'book' is simply a pointer to a folder. So is a SmartPath and a DropDown List. Therefore, once you have set a SmartPath, shelved a book or created a DropDown List, you can easily navigate to the source folder by simply taking advantage of the pointer. You no longer have to navigate to it.

The ease by which Pathagoras allows your access to your source clauses is surpassed by no program. No document stored within a Pathagoras control is more that one or two clicks away.

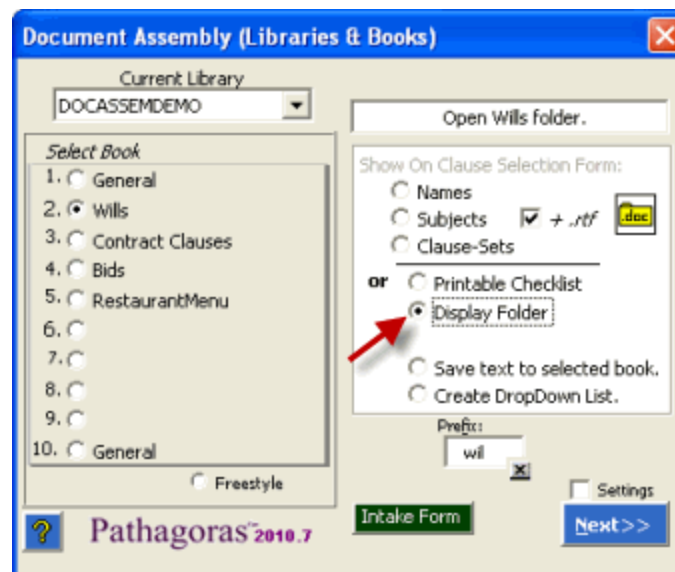
Being aware of the tools illustrated below will save you and your staff countless time daily (not to mention eliminating the frustration of trying to remember where in the computer or on the network things are located.)

 Remember that when you are assembling a document, you have only displayed a *copy* of the document, not the original. Whenever you want to edit the *original* text, or to add or delete files from the source folder, follow the steps outlined below to quickly access the folder.

33.1 Locating document in a Book

The easiest way to get to the folder of a 'shelved' book is via the initial Pathagoras Document Assembly screen.

1. Click the Document Assembly icon. The Libraries and Books screen will appear.
2. Click once on the book you wish to display. The screen will expand.



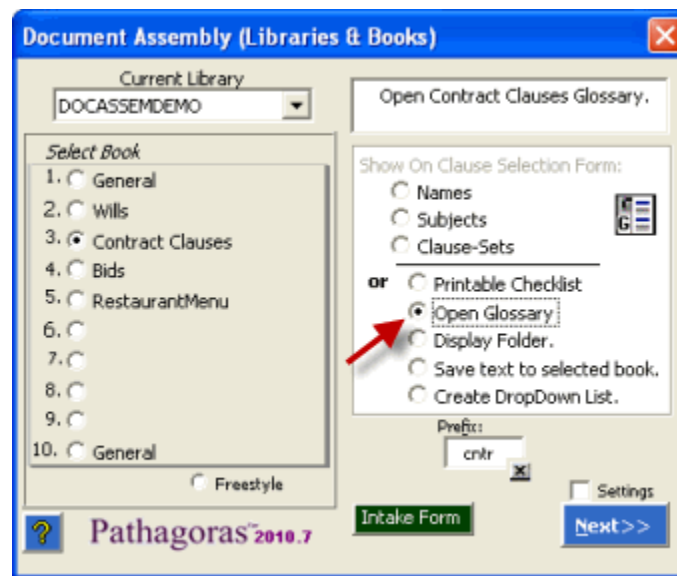
3. Click the "Display Folder" option.
4. Click the Next button and the target folder will display. (Double clicking the "Display Folder" button will combine steps 3 & 4.)

33.2 Locating clause in a Glossary

There are two basic methods you can use to access terms in a glossary for editing:

Depending upon the size of the glossary, the fastest way is probably via the initial Pathagoras Document Assembly screen.

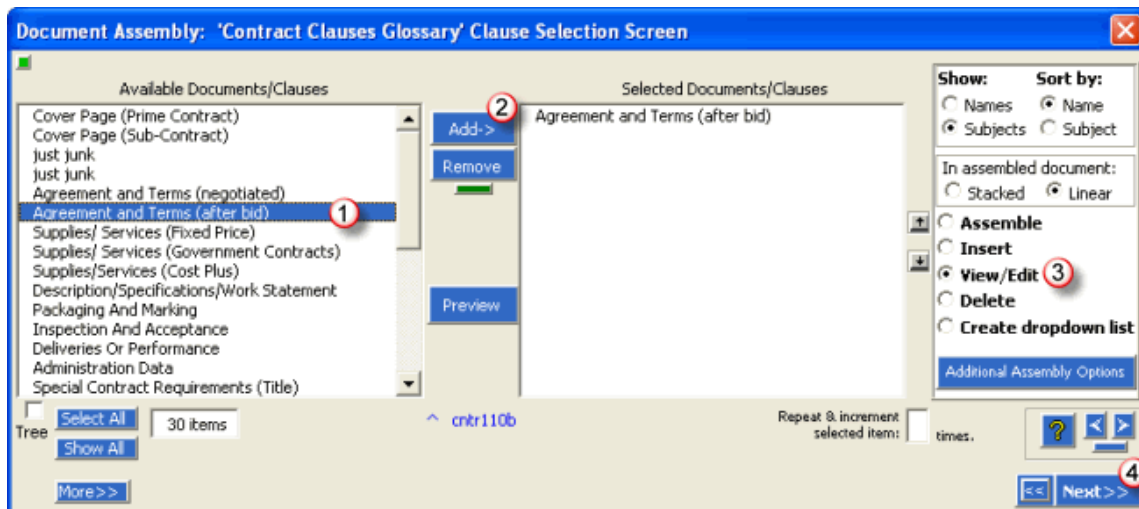
1. Click the Document Assembly icon. The Libraries and Books screen will appear.
2. Click once on the glossary you wish to display. The screen will expand.



3. Click the "Open Glossary" option.
4. Click the Next button and the target folder will display. (Double clicking the "Open Glossary" button will combine steps 3 & 4.)
5. When the glossary appears, simply scroll to the clause you wish to edit.

If the glossary is large, it may be faster to take this route:

1. click the Document Assembly icon. The Libraries and Books screen will appear.
2. Click book you wish to display. From the expanded screen, click the Next button to display the clause selection screen. (Double clicking on the option button reflecting the glossary achieve the same result.)



From the Clause Section Screen:

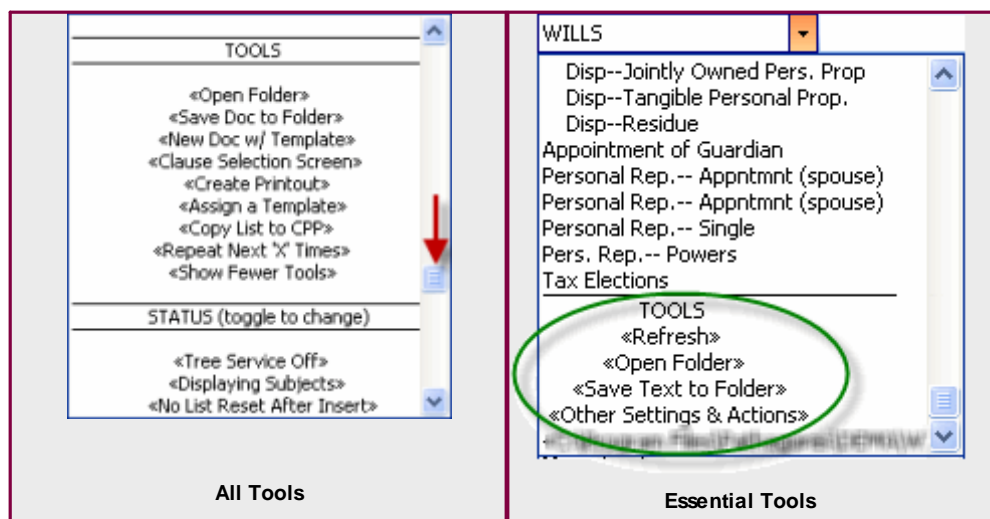
1. Select the clause you wish to edit from the left panel.
2. Click the Add button to move it to the right panel (Double clicking on the desired clause in the left panel will achieve the same result as 1 & 2.)
3. Select the "View/Edit" option from the action panel.
4. Click Next. (Double clicking "View/Edit" will achieve the same result as 3& 4.)

Pathagoras will locate and open the glossary and highlight the target term. Edit as desired.

33.3 Locate a clause in a DropDown List

To display the source folder of a term in a DropDown List (so that you can edit an existing term or to add or delete terms), just click down the DropDown List itself.

At the bottom of each List are a series of tools. Depending upon whether you are displaying "All" tools (figure 1) or just the "Essential" tools (figure 2) it will appear in the first or second position.

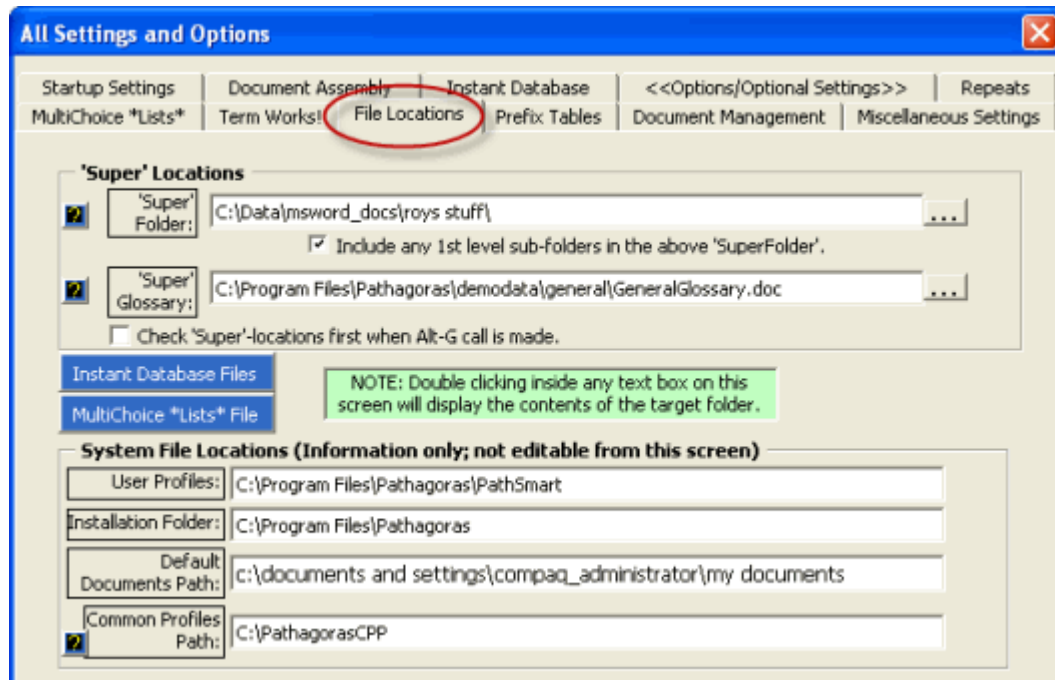


If the DropDown List represents the contents of a folder, the target element is called

<<Open Folder>>. If the DropDown List represents the contents of a glossary, the target element is called <<Open Glossary>>. Click the appropriate element and the desired action will take place.

33.4 Text boxes displaying folders

On any number of screens in the Utilities/Settings (see in particular, "File Locations" which we illustrate below) and other sections of Pathagoras you will see text boxes which display references to folders in which substantive or administrative files are stored. You can quickly navigate to those folders simply by double clicking inside of the text box.



The Pathagoras System

Compatibility Issues

Part



34 Compatibility Issues

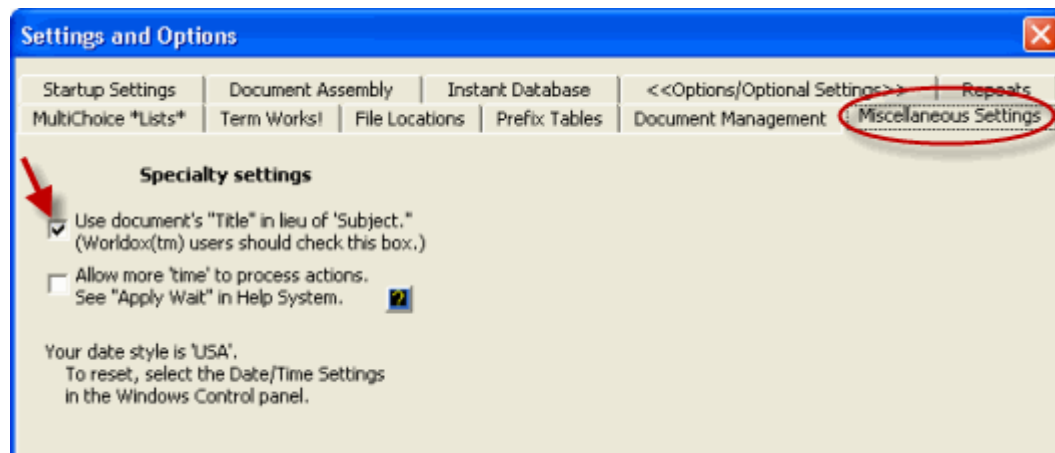
Pathagoras is a Microsoft Word add-in. It is programmed in VBA (Visual Basic for Applications). It is compatible with Word2000, XP, 2003, 2007 & 2010 (32 bit only). (Support is no longer offered for Word98.) Pathagoras is compatible with all versions of Windows.

All known compatibility issues with other programs are discussed in the following sections.

34.1 Wordox®

Pathagoras Document Assembly is fully compatible with Wordox®. However, Wordox captures the 'Subject' property in the built-in Document Properties (File|Properties). Pathagoras adapts to this by using the otherwise unused 'Title' property. If you use Wordox for document management, you should advise Pathagoras to use this property by performing the following steps:

1. Go to the Utilities/Settings screen.
2. Click the 'All Settings' button. You will be taken to that screen.
3. Select the 'Miscellaneous' tab.
4. Check the 'Use Document's Title in lieu of Subject' box.
5. From now on, the Subject will be stored in (and read from) the document's 'title' field, preserving the 'subject' for Wordox.



Two additional matters concerning Wordox.

1. By default, Wordox will assign a cryptic name to your documents. You can, however, override that assignment with a name that you personally assign. It is absolutely possible to adopt Pathagoras [prefix/suffix naming convention](#)⁴²⁰ while using Wordox, and to use Alt-G to instantly recall a document stored by Wordox.
2. As you are creating, Pathagorizing and testing your documents and your entire document assembly system, if Wordox is simultaneously active, it will dramatically (and 'frustratingly') interrupt your editing process every time you try to save a document. We strongly recommend that you disable (but only temporarily; just while editing) Wordox while you are building and testing your system.

When done, let Wordox back in to do its thing with regard to your newly created

documents. (Alternatively, build your Pathagoras system on a computer that does not have Worldox loaded. You can point Pathagoras to your non- Worldox managed source documents, using Worldox to manage your 'final' documents, or you can transfer your completed documents back over to a Worldox managed computer where they will be integrated into your Worldox system.)

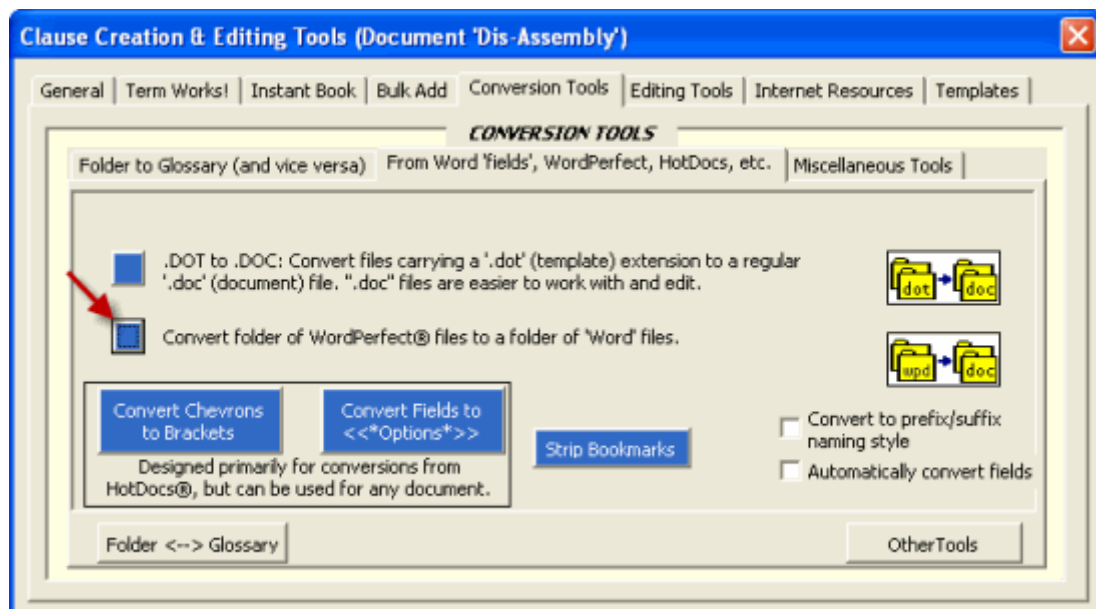
Worldox® is a registered trademark of the World Software Corporation
--

34.2 WordPerfect®

Pathagoras does not come in a WordPerfect® version. Click [here](#) for explanation.

That said, there are quite a few ways that Pathagoras 'plays nice' with (and actually can automate) your WordPerfect documents.

- Pathagoras let's you assign a folder of WordPerfect documents to a [DropDown List](#)³⁷⁸. So, you can point and click in any WordPerfect document into your Word editing screen quickly and without navigation as if it were a native Word document. The 'how to' for this is shown further down this page.
- Pathagoras lets you assign a folder of WordPerfect documents to a Pathagoras book. With that book, you can create a 'Clause Selection Screen' which displays all of the documents in that folder. Select and assemble one, several or all of the documents. The kicker: if the source WordPerfect document(s) contain plain-text [bracketed] variables, those variables will be usable in Pathagoras with no further editing.
- Pathagoras can convert your WordPerfect documents *en masse* into Word documents. To do so, drop down the Pathagoras features menu and choose "Clause Creation Tools." Click the 'Conversion Tools' tab and then click <Convert folder of WordPerfect files to Word>. You will be asked to navigate to the folder containing your WordPerfect documents. The conversion will take place automatically.



- You can easily take text from any WordPerfect document and convert it into a clause for use in your Pathagoras system. Simply copy it from, paste it into a blank Word document. Highlight it (or the desired part of it) and press <Alt-G>. This will activate [TermWorks](#) ³¹¹. From there you can add the text into any existing Pathagoras book.

DropDown Lists with WordPerfect content:

Because Word can convert a WordPerfect® document into a Word document, so can Pathagoras. The trick to document assembly with WordPerfect documents is to use **DropDown Lists** as your gateway to your WordPerfect documents. (As a reminder, a DropDown Lists is an always on, always active listing of the content of a designated folder. You can assign any folder to one of up to 10 DropDown Lists. 'Content' includes WordPerfect documents.)

Follow these steps to assign a folder containing WordPerfect documents to a DropDown List:

1. Click the DropDown Lists item in the Pathagoras features menu. The following screen (Figure 1) will appear:

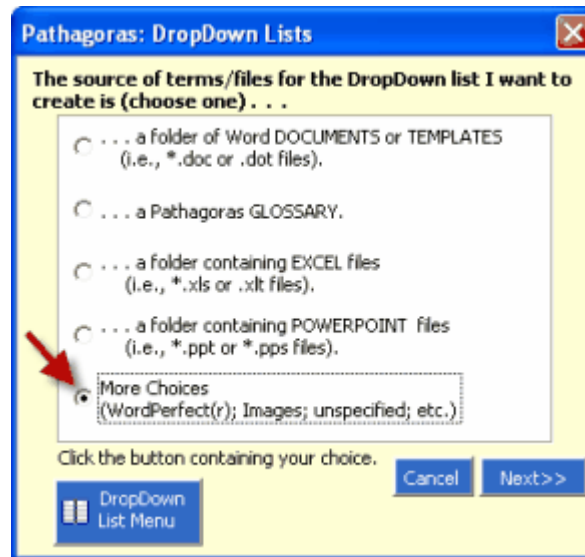


Figure 1

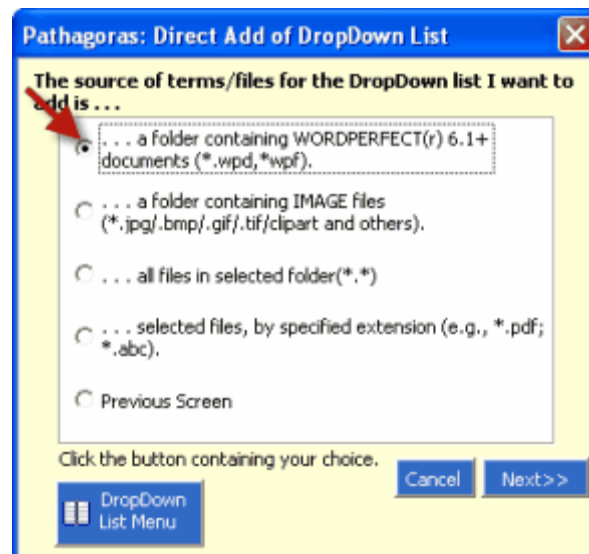


Figure 2

2. Click the "More Choices" option and Figure 2 will appear. Click the first option.
3. Follow the prompts to navigate to the folder containing the WordPerfect files. When there, select a single file (it doesn't matter which one -- it is the folder that is being registered, not the selected file -- and click 'Ok.'
4. Provide a name for the List. That is all there is to it. Now you can point and click your WordPerfect files into your document without having to ever navigate to the folder again.

If you wish to select multiple items from the DropDown List, you can quickly convert the list to a Clause Selection Screen by selecting that entry from the 'below the line' options. See this link: [WordPerfect Assembly](#) ³⁷⁸

NOTE: While the source is WordPerfect, the text and the resulting document is Word.

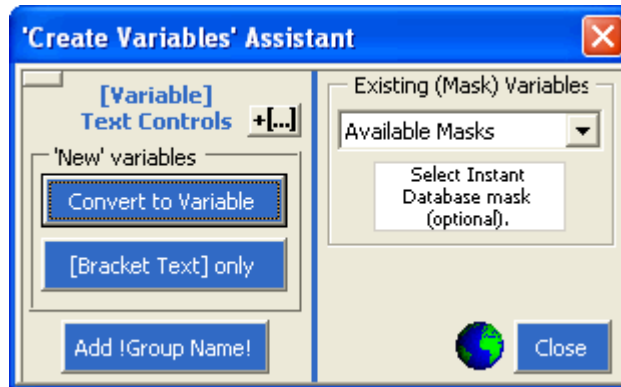
34.3 HotDocs®

Pathagoras cannot work directly on HotDocs® documents. They are quite incompatible programs. HotDocs requires hidden fields, ancillary documents and complex code sets. Pathagoras requires nothing of the kind.

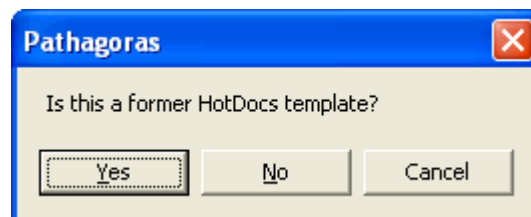
But what if you have a large library of HotDocs documents. Are they useable at all? We are pleased to say 'yes.' Pathagoras can convert many of the variables that reside in a HotDoc-ed document into plain text bracketed variables. Such ability will allow you to either partially or totally convert your HotDocs library into a Pathagorized documents.

The conversion process is remarkably simple and starts with Pathagoras' standard 'Create Variables' Assistant. Here are the steps:

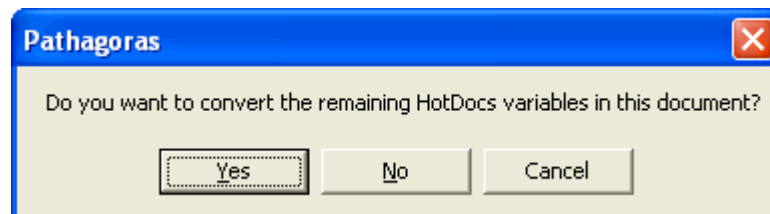
1. Call up a HotDocs document (Word or rtf).
2. Highlight one of the variables. Be sure to include both the opening and closing chevrons (the double angle brackets that surround the variable).
3. Click Alt-V to bring up Pathagoras 'Create Variables Assistant'.



4. Click the button that reads Convert to Variable. Pathagoras will recognize the highlighted term as a possible HotDocs variable (and therefore the document as a former HotDocs document.and ask for your confirmation.



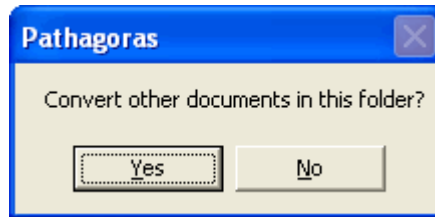
5. If you click 'Yes', Pathagoras will convert all instances of that particular variable to a plain text bracketed variable, and remove all invisible field and links. Pathagoras will then ask . . .



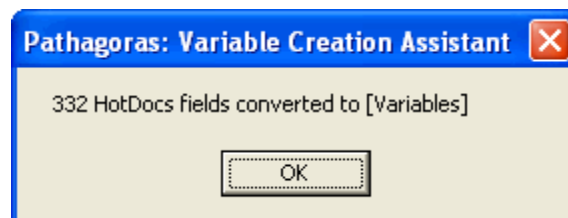
6. . . . to which you should answer 'Yes' if that is the result you desire. Pathagoras will


then convert the remaining HotDocs variables in the open document.

7. Once the HotDocs variables in the current document have been converted to plain text variables, Pathagoras will next ask . . .



8. . . .to which you might answer 'Yes' if that is the action you desire. Pathagoras will then convert the HotDocs variables in all other documents in the current folder. When all documents have been processed, a confirmation screen like the following will appear:



 Because Pathagoras can convert all files in a folder, it would be a good idea to pour all related HotDocs documents into a (new) folder before beginning the conversion process. That should save you a lot of time.

Limitations: While we are hoping to develop (and are currently working on) routines that will convert even the most complex behind-the-scenes formulae that can reside in HotDocs variables, Pathagoras currently can convert only simple HotDocs variables (which probably constitute 99% of what we have seen thus far). If you have fields that did not convert as you wished them to be, forward them on to us so we can add them to our 'conversion wish list'. We'll let you know if and when we have succeeded.

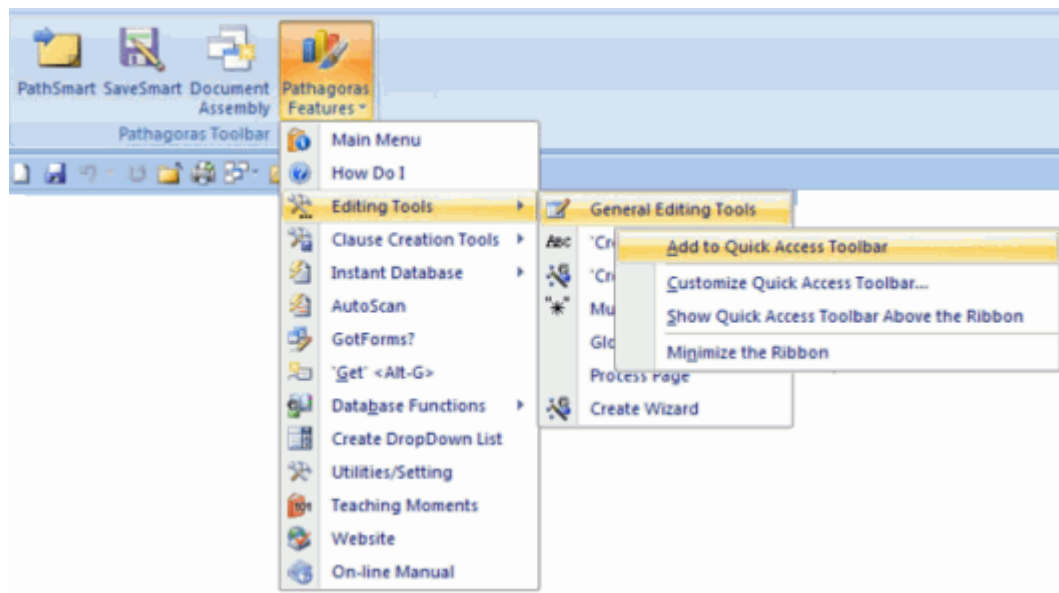
35 Quick Access Toolbar

You can run all Pathagoras features from the buttons and boxes behind the Pathagoras tab.

But as a practical matter, it still will take at least two or three clicks to gain access to most features. These steps below will allow you *one-click access* to the features and tools you most commonly use, and to create a highly personalized toolbar.

The "QAT" (*Quick Access Toolbar*) is a special toolbar that Word 2007/2010 provides which allows you one-click access to any pre-programmed action. The QAT that ships with Word 2007/2010 is initially populated with the "Save", "Undo" and "Redo" commands. You can add any other Word commands to the QAT. Do so by right clicking on any button and select the "Add to Quick Access Toolbar" element in the list.

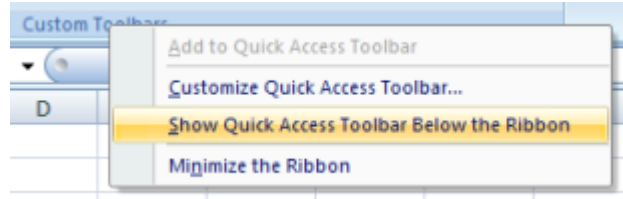
Any feature or button in Pathagoras can be placed on the QAT in the identical fashion. Simply display and then right click on the feature that you want added to the QAT. (It can be any button or menu that you see, whether topmost or bottommost of in the middle.) The same icon as appears in the main menu area will be copied to the QAT.



You may wish to place Pathagoras functions in a group and set out the group with a 'spacer.' This can easily be accomplished. Click the option called "Customize Quick Access Toolbar." There you can rearrange the functions that current reside in the QAT and add the appropriate spacers to your liking.


Position of QAT:

On other thing to note is the position of the QAT relative to other tabs and ribbons. In the first picture above, the QAT is above the Ribbon. In the second picture, it's below the Ribbon. It is a matter of personal preference, but if you like having favorite features closer to the editing screen, you can change the position of the QAT. As shown in the third picture, below, just right-click on the QAT and select the option to show it below the Ribbon.



With the QAT below the Ribbon, you may lose a little bit of screen real estate, but you make up for it with added convenience and increased productivity of having your 'favorites' within easy reach.

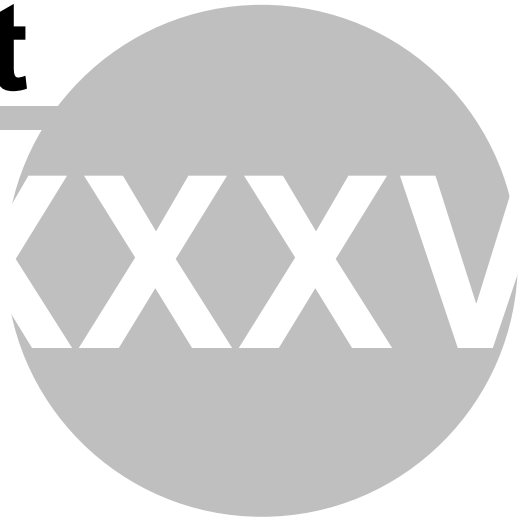
When assigned to the Quick Access Toolbar, the Pathagoras icon is represented by a set of stacked folders.

 Adding (and removing) elements to (and from) the QAT could not be easier. So don't hesitate to treat the QAT as a fluid tool. When you are Pathagorizing documents, add the Pathagorizing tools you like the most to the QAT. When you are done, remove them. Add other tools to the QAT as you are using them intensely, and remove them when you are done.

The Pathagoras System

Excel Connections

Part



36 Excel Connections

A stand-alone 'Pathagoras for Excel' module is available which allows you to build some pretty sophisticated (and simple 'pretty' intake forms which you can send to your client or customer for completion. When returned to you, the Excel module allowed you to click a button to create an IDB record for Word. Other features are also available in that module. See this page in the Manual. (A link to the *Pathagoras for Excel* stand-alone can also be found at that link.)

Beginning with version 2013.1, Pathagoras allows some pretty strong connectivity with your Excel spreadsheets within the Pathagoras module itself. Click [here](#) to be taken to the page of this Manual that discusses ["Excel and the Instant Database"](#).

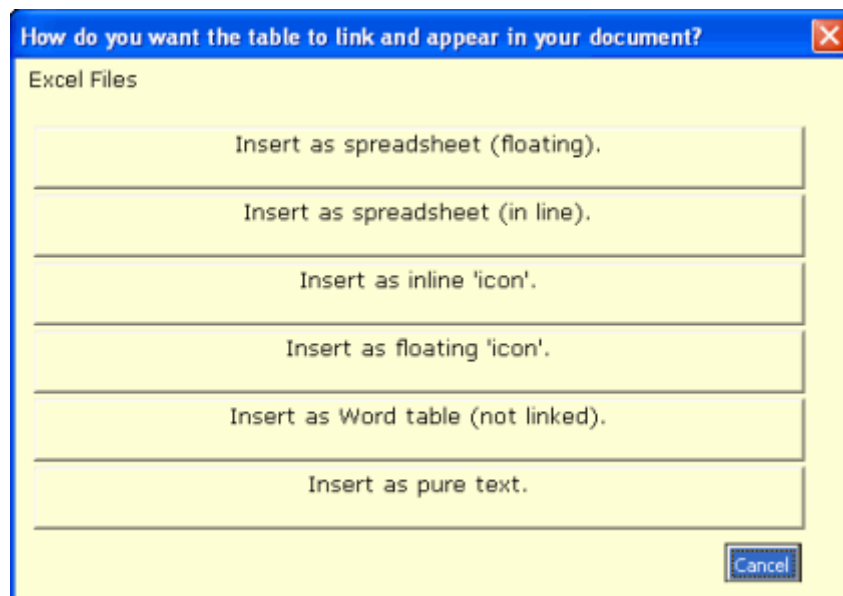
36.1 Excel and DropDown Lists

Pathagoras* allows several ways for you to access data stored in Excel spreadsheet.

Via DropDown Lists:

- Import entire spreadsheet directly into document.
- Import named range of cells directly into document either as linked spreadsheet or as a Word table.

To enable these features, assign one of your DropDown Lists to a folder containing Excel spreadsheets. When you click on a spreadsheet you want to import, you will be provided these options:



If one of the first two choices is made:

- Pathagoras will ask you if you want the content linked or not linked. When linked, changes you make will be reflect back into the source spreadsheet. Not linked will

not.

The selected spreadsheet (or named range) will appear inside your document as a normal spreadsheet. It will show column headings and row numbers. You can edit it as if it were a 'real' spreadsheet (because it is), and you can perform normal spreadsheet functions (math, totals, etc.)

- When 'Insert . . .floating' is selected, the spreadsheet can be dragged and dropped and resized with your mouse.
- When 'Insert . . .in line' is selected, the spreadsheet takes on characteristics of text. You can copy, paste and move like a text block.

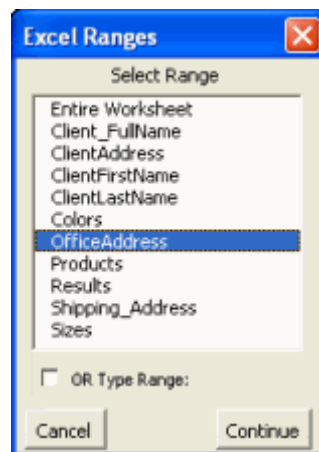
If you select 'Insert as icon', the actual spreadsheet does not display. Rather, a picture with a link to the spreadsheet is inserted. The icon is double-clicked, will open the selected spreadsheet in Excel.

When 'Insert as Word table', an actual spreadsheet is not inserted. Rather a Word table of the same dimensions of the spreadsheet or range of the spreadsheet is created and the individual cells are filled with the content of the original cells.

When inserted as plain text, no table is created. Just the text of the spreadsheet or the range is typed into the document. A tab will be inserted for content in a different cell on the same row, and an 'enter' will be inserted for content beginning in a new row.


Named or Hard Coded Ranges

After you select a button from the above screen, Pathagoras will open the spreadsheet. It will determine if there are named ranges within the spreadsheet. If there are, Pathagoras will present those ranges to you.



- If you want to insert a named range, click on the name.
- If you want the entire spreadsheet, click the item that appears at the top of the list.
- If you want to 'hard code' a range, click the "Type Range" checkbox toward the bottom of the screen. A textbox will appear that will allow you to designate the range of cells you want to import. Type the cell designators for the upper left and lower right corners of the range. (Be sure to designate a cell using the proper cell-naming style, e.g., 'A1', 'C22', etc.)

Click the Continue button to proceed.

 The normal effect of the "Insert/NewDoc" toggle in the DropDown List area still applies. If you want the information inserted in the current document, make sure the toggle is set to 'Insert'.

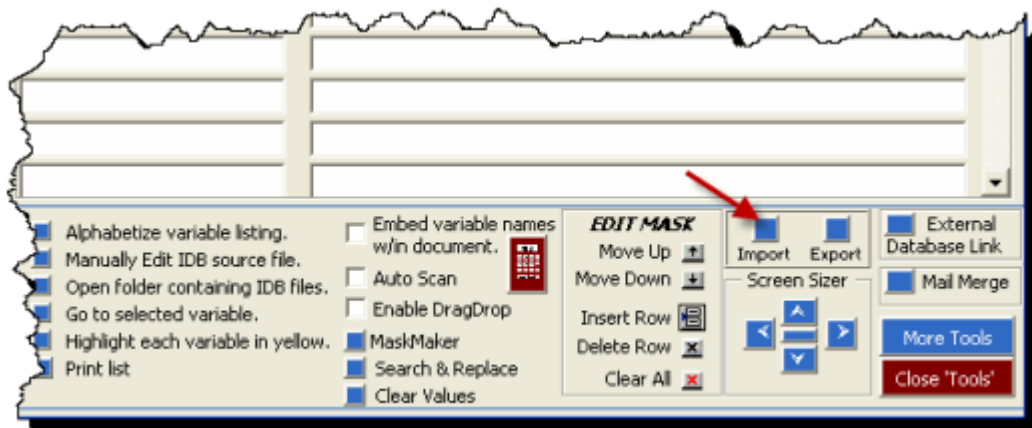
*Several features described herein are available only with Pathagoras Version 2013.1+

36.2 Excel and the Instant Database

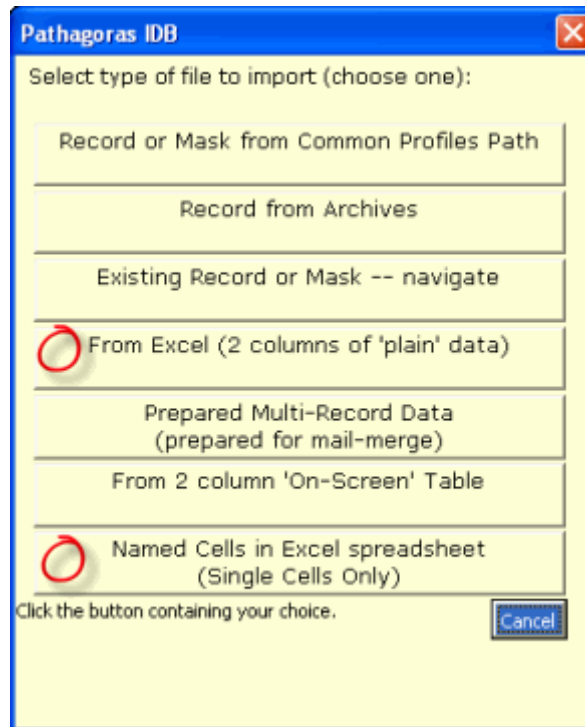
Pathagoras allows several ways for you to access data stored in Excel spreadsheets directly from Word.

Importing Data to the Instant Database screen:

- If you want to use data that currently exists in an Excel spreadsheet (perhaps one that contains all relevant information regarding a client or customer, you can easily import that data directly in an Instant Database record. First, display the Instant Database screen (typically <Alt-D>) and click the red Power Tools button. Click the Import button on the resulting screen.



Pathagoras will offer you the following choices:



For this topic, we are interested in just two items in the list*

- Import two column Excel spreadsheet directly into Instant Database.

Perhaps you emailed to a client a simple spreadsheet, column #1 containing field names and column #2 initially blank. Your client filled out column 2 and sent it back to you. Pathagoras can easily read that spreadsheet and place the information onto your Instant Database screen. Click the 'From Excel (2 columns of 'plain' data)' item. Following the prompts, navigate to the file and Pathagoras quickly fills in the IDB form. Proceed as normal.

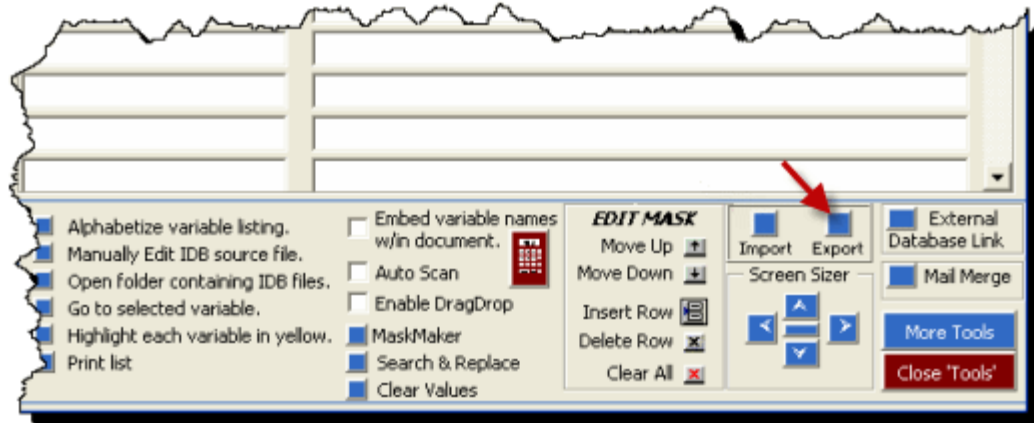
(To learn how to create that simple Excel spreadsheet that you can mail to your clients/ customers, see below section titled "Exporting Data from Instant Database Screen.")

- Import named cells in Excel spreadsheet directly into Instant Database.

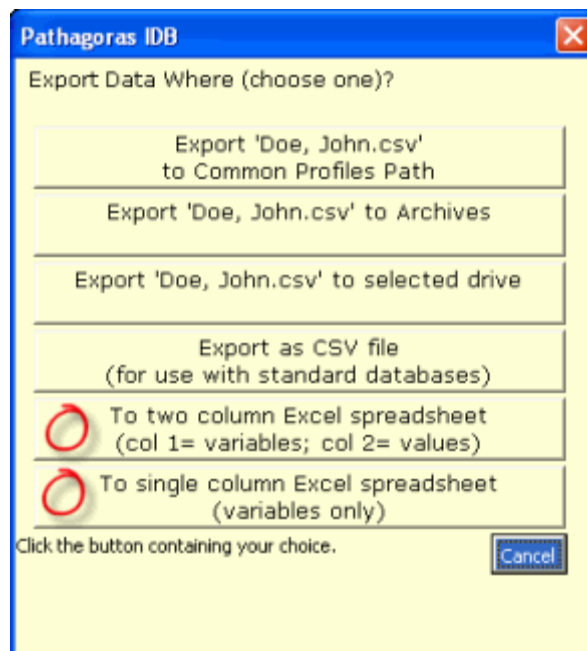
Perhaps you have a 'fancy' spreadsheet containing lots of data pertaining to your clients case. And perhaps you have assigned 'names' to various cells in the document that contain data that you wish to use in Pathagoras. It is easy to access each named range and import it into your IDB form. Click 'Named Cells in Excel spreadsheet'. Pathagoras will quickly find each Named Cell in the spreadsheet and return its value onto the open IDB form. If you want to

- With above tools, it is easy for you to design in Excel an intake form (simple to fancy), and send that form to your client or customer for completion. On its return, quickly import the data into your records or into your document. No reentry of data required.

Exporting Data from the Instant Database screen:



The following screen is produced:



For this topic, we are interested in just two items in the list:

- Export the data to a two column Excel spreadsheet.

Perhaps you want to email to a client a spreadsheet showing the current state of the data you have regarding that client. Instruct the client to peruse the spreadsheet, to make any corrections as appropriate, and to send it back with corrections.

- Export the data to a single column Excel spreadsheet..

Perhaps you want to email to a client a *blank* spreadsheet showing only the list of variables in the left most column. Instruct you client to fill out the column to the immediate right and sent it back. Then you can.

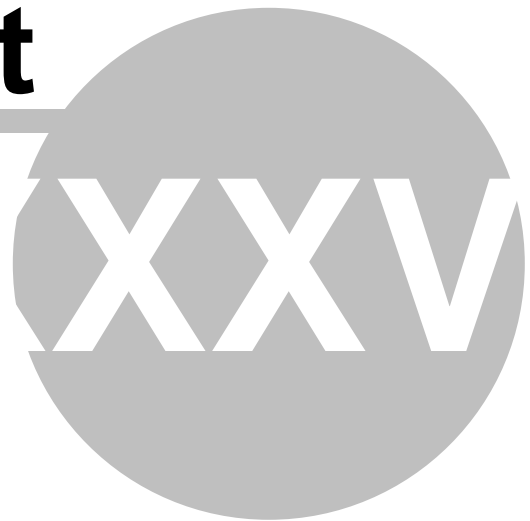
* Several features described herein are available only with Pathagoras Versions 2013.1 and later

** To learn more about 'Prepared Data (Mail Merge)', [click here](#)  300.

The Pathagoras System

Comparative Approaches

Part



37 Comparative Approaches

For most activities, Pathagoras provides at least two methods to achieve a desired result. Sometimes there are 3 or 4 alternatives. However, there will never be a situation that all options will implemented by any one user. Indeed, many users may never know that an optional approach is even possible. If you like the way you are doing things, stick with it.

But it should be comforting to know that shortcuts, hot-keys and alternative approaches are available for just about every given task. To the extent you are not wedded to one approach, come to this section and learn about others. That way, you can make a more informed choice on which approach may be best for your office.

Some of what is discussed here repeats or summarizes what is stated elsewhere. A link to the appropriate 'detail section' of the Manual is provided for each section below.

37.1 Clauses vs. Templates

Clauses are snippets of text designed to be called into a document under construction 'one at a time.' Assembly is typically accomplished via the Clause Selection Screen.	Templates are 'complete' documents. They typically contain automated <<*Optional*>> text blocks that allow the user to choose whether that block of text should be kept or deleted.
--	---

A major decision that you, as the systems designer, will have to make in creating your first document assembly system is whether to build the system based on a 'clause assembly' model or a 'template' model. The text in the below section will help guide you in making that decision.

But as you read the below text, keep in mind that it is really not an 'all-or-nothing' decision. Our approach in designing Pathagoras was to provide you as much flexibility in system design as possible. You can mix the two approaches and easily go back and forth as you need to and learn more about how your end users will be using the source document. And if you commit to one approach and decide to adopt the other, not to worry. Pathagoras provides the tools needed to make the switch.

To the extent that we can recommend something to you, we encourage, and have built our demo systems around, the 'clause assembly' method, with judicious use of Options and Optional blocks (a 'templates' feature) within many of the clauses.

Pathagoras offers two distinct approaches to document assembly. Despite the title to this section Clauses VS Templates, the approaches are not opposites. Rather, they can complement each other quite nicely when you fully understand the strengths and weaknesses of both.

At the outset, you will make some decisions as to the basic approach you want to take. However, those decisions are never etched in stone. Pathagoras makes it easy for you to change your mind.

Let's discuss the two basic approaches.

Document Assembly by Clauses: This connotes 'paragraph assembly.' You would maintain a collection of text snippets, each saved as individual documents in a particular folder. Each snippet would typically represent a 'stand-alone' section of a larger whole.

- Sometimes the snippet is a mandatory part of the document;
- Sometimes the snippet is an optional clause. It will be called into some documents, but not others.
- Sometimes the snippet is one of several alternative clause. One of them will always be called in, but not necessarily all of them. These snippets might be labeled "ChoiceA", ChoiceB, ChoiceC.

Collectively, the snippets represent all of the possible text choices regarding a particular type of document.

In a typical document assembly session, a user would select the book that contains all of the clauses. (A book is simply a pointer to a folder that you have designated.) Pathagoras reads the contents of the folder and displays all of the available snippets (i.e., documents in the folder) onto the 'Clause Selection Screen' (in the left panel). The user would select the clauses needed for the particular client/customer, moving the clauses from the left panel to the right. When the proper clauses are at the right, and in the proper order, the user clicks Next>> and the selected clauses are instantly assembled into a new, unique document.

Keep in mind that what is being displayed in the Clause Selection Screen are the documents in a folder. (Pathagoras doesn't know that they are just 'snippets' of text.) The beauty (and elegance) of such a system is that if you add more snippets to the folder, more snippets will display on the Clause Selection Screen.

Advantages of the 'clause based' approach:

- it is easier to effect a change that affects multiple documents. Let's say that you moved your office. Let's further say that your office's address appears in the signature block of every single document that you create. If you assemble documents, calling in the signature block as a snippet at the end of each new document, you only have to change the one snippet and it permeates all future documents built with that term. (The alternative, of course, is to open each 'complete' document/template and make the address change in each.)
- when you need just a single clause, it is easy to point and click it in. Of course, if you don't have a [DropDown List](#)^[356] of the clauses, you really should.
- you can call in a single clause simply by typing its name to the editing screen and pressing <Alt-G>. Depending upon how the clause is named and where it is stored, Pathagoras can typically find the clause almost immediately and insert it into your document with your having to navigate to find it.
 - If the clauses are named following the [prefix/suffix naming style](#)^[420], and the clause will be called directly from the folder or glossary which has been associated with the prefix. Response time is practically immediate.
 - if the clause has been saved in one of your [SuperBooks](#)^[90], Pathagoras will hunt it down there, again practically immediately.

[Click here](#)^[330] to be taken to the section of this Manual that discusses how to '[dis-assemble](#)'^[330] a currently 'complete' document into its component parts. This will serve as the starting point for a 'clause assembly' system.

[Click here](#)⁴⁰⁶ to be taken to the section of this Manual that discusses how to create [Clause Sets](#)⁴⁰⁶. (Clause Sets allow the end user to create a complete document by selecting a single 'set' of pre-designated clause snippets. Clause Sets allow you to truly have the best of both worlds.)

Document Assembly by Templates: A template contemplates a complete (actually, overbuilt) document. It typically had all, or at least most of the text that a standard (and perhaps super-standard) document of its type would contain. Variables (placeholders for personal information, addresses, dates, etc.) are strategically placed throughout the document. During the document assembly session, the end user will provide the personal information. Frequently a template will be overbuilt. It will contain text sometimes stays in, or sometimes is deleted, depending upon the specific client/customer circumstances or desires. Those text blocks can be marked as <<*Optional*>>. Sometimes the text block should reflect choices. Those can be marked as <<*Options*>> blocks.

Advantages of the 'template' approach:

- It is conceptually easier. There is only one new concept being introduced to the document assembly beginner -- the concept of variables that need to be completed to end up with a final product. It simply is easier to learn one new thing at a time. (So the suggestion to the system administrator is this. 'Pathagorize' a template and use it to introduce the users to the concept of document assembly as variable completion. Later, dis-assemble the template into its component parts and see if 'clause assembly' might work better for your office.)
- For the document assembly administrator, template assembly is likewise quite a bit easier to implement. You don't have to 'disassemble' a document into its component parts. (While we recommend that you eventually adopt the paragraph assembly approach, it is not necessary at the outset. Document '*dis-assembly*' can be postponed until a time when you and the office is 'ready' for that new dimension.)
- Formatting and style management can be more easily controlled within a single document. If your office uses (either intentionally or inadvertently) mixed styles, you may get unexpected results when you assembly clauses by snippets. The proper 'solution' is to make the styles uniform across all snippets and templates, but the 'advantage' of templates is that you don't have to mess with this right now. You just need to make the particular template internally consistent.
- Since templates are complete documents, the end user initially sees a result that is more complete and perhaps more intellectually satisfying.
- Other document assembly systems tend to favor this approach. Therefore, it is perceived to be an industry 'standard'. And if you have tried another system, that will be your default 'assumption' and bias. That is perfectly okay, but to the extent we can begin moving you over to the concept of 'snippets,' you can dramatically improve document assembly efficiency.

Pathagoras and the 'Industry Standard'

The industry standard for document assembly source document is 'template based.' Pathagoras allows, but does not promote, that standard. The reason why Pathagoras does not embrace the industry standard is that the standard has not

worked. Document assembly as an office 'product' has been around for more than 15 years. Very few people and offices use it, even though 10s of thousands of copies of Pathagoras' competitor's program have been sold.

In the long run it is tougher and more confusing to try to push an entire document, which run the gamut from simple to complex, into a template. Templates can become exceedingly long, complex and time consuming to write. That is why so much document assembly 'software' has become document assembly 'shelf-ware.' Administrators were finding that they couldn't write the logic blocks accurately. End users were overwhelmed by having to learn a new program. The program was shelved until more time could be found to implement it.

With paragraph assembly, the clauses are still selected by the end user, but the administrator does not have to be a programmer. Each separate clause is self-contained. No options text to type, not programming-like code to create. Just simple plain text clauses that assemble quickly and accurately.

Notes: Except for the fact that a 'paragraph' is typically smaller than a 'template,' there is no technical difference between the two concepts. They are both 'documents' as far as Pathagoras or any user can tell. Each can contain [bracketed variables]. Each can contain <<Options>> and <Optional>> text blocks. Each can contain references to other clauses (set out between "<<" and ">>" markers.)

The main difference between a 'clause snippet' source file and a 'template' source file is the average size of the file. Snippets are intentionally short. They are designed with sequentially pasting of one clause after another one in mind. Templates are intentionally long. They are designed with presenting the whole, and 'keeping' or 'deleting' optional text in mind.

So, keep this in mind. No matter how complete the template might be, it likely won't contain *every* possible option within its four corners. So you will likely have a need for at least a small library of clause snippets that you can quickly reference for that additional text needed to put on the final touches. And not every clause snippet should be free from options and optional text blocks which are typically associated with templates. Use them when you need them.

37.2 Naming Your Clauses

37.2.1 Prefix/suffix style vs. descriptive names

The prefix/suffix naming style allows instant recall of a clause. Just type its name to the screen and press <Alt-G>.

The clauses 'subject' can be stored as a document property and quickly displayed as an alternative to the 'name' display. That way, the end user can quickly see the

Descriptive names is the more 'natural' naming style. Users are used to this. Many may resist a change to a prefix/suffix naming style until the benefits are demonstrated and understood.

clause's purpose.	
-------------------	--

If you adopt the prefix/suffix naming convention for your clauses or terms, you will be able to recall any document or term from any source instantly. And this is regardless of which glossary or which path in which the term is stored. This feature alone should convince you to adopt the prefix/suffix naming convention. But read on for some comfort words.

Pathagoras recognizes that a term bearing a name such as "cntr219a" is, at least to the inexperienced user, gibberish. A way to provide a meaningful description for the clause is essential as well. That is why Pathagoras lets you assign a "Subject" to each document that you save in the prefix/suffix style. With the subject always available and in view, the end-user will always be able to make an appropriate selection.

As you become more adept with using Pathagoras, and realize the benefits of 'mouseless' assembly using the prefix/suffix naming convention, you may wish to convert existing documents to this protocol. You can easily convert an entire folder's worth of documents from descriptive names to 'prefix/suffix' names. The entire process can be easily accomplished using Pathagoras' exclusive [Names & Subjects Editor](#).

You can print a listing of your clause names (in the prefix/suffix style) along side of their subjects. This serves as a handy reference and can be used as a checksheet for composing a document from individual clauses. To generate such a list, display the Document Assembly Libraries & Books screen. Choose the appropriate book and then click the "Printable Checklist" item found in the center panel. Click here for illustrated instructions: [Checklist](#).

Read more about this naming style here: [Prefix/Suffix Naming Convention](#).

For instructions on how to assign a prefix to a folder or glossary, click here: [Prefix Table](#).

37.2.2 Manual vs. Auto Naming of final documents

If your office has good 'discipline' in consistently naming client/customer documents, continue using what works.	If your office does not have good 'discipline' in consistently naming client/customer documents, consider this SaveSmart tool.
---	--

The previous discussions concerning 'naming conventions' dealt with the names you assign to source clauses for your document assembly system. But once you assemble a document for a client or customer, you are presented with another naming issue. If you save one letter "Letter to Mr. Jones re Smith" and another letter "Smith Letter to Mr. Jones," you also have a naming problem. The problem will be most evident when you need to recall the letter—"Now what did I call it?" If you do not now have a document naming convention, you need to develop one.

The SaveSmart function offers a method that will lead to more uniformity in naming conventions. You can implement the routine, as establish the naming parameters by clicking the black button to the left of the "Propose Name?" checkbox at the very bottom of the SaveSmart screen.

37.3 Naming Your Variables

37.3.1 Bracketed vs. Non-bracketed variables

A bracketed variable is simply any text that you intend to be a variable that has been surrounded by brackets. E.g., [Customer Name].

Despite the suggestions and myriad of examples in this Help System, Pathagoras does not mandate [bracketed variables]. Any text can be a variable. “Customer” is just as good as “[Customer]” in many cases.

Of course, if the word you select as a variable might otherwise appear naturally within the document, then that word should not be used as a variable.

- ‘Customer’ as a variable is ill advised if the word ‘Customer’ appears in the document as text that you want to keep intact. (E.g., “The Customer shall be responsible for all shipping costs.”) When Pathagoras searches out ‘Customer’ for replacement with person data, you will end up with “The John Doe shall be responsible”)

So, in order to segregate a description variable from other occurrences of the word in the document, brackets are certainly a good way to go. “[Customer]” is not the same as “Customer”.

Brackets also make the variable stand out in the document. As you are perusing your document for its variables, it certainly easier to see [Customer] than Customer.

If you plan to take advantage of the <Scan> feature associated with Pathagoras Instant Database and *GotForms?* modules, the brackets are mandatory. <Scan> searches the active document for instances of bracketed variables and presents them to you for replacement.

While on the subject of brackets, if you decide to use them, please know that Pathagoras does not care what kind of ‘brackets’ you use to identify your variables.

- You can use [square] [brackets] (the default) or any other type of enclosing character.
- So ‘(’ and ‘)’ would work. However, since parentheses occur with great frequency in standard documents when no variable is intended, it is probably not a great idea to use parenthesis.
- But “(“*” and “*)” could work, as could several dozen other single and double character combinations. ‘{’ and ‘}’ or ‘<’ and ‘>’ or ‘#^’ and ‘^#’.

Rules (just two):

- the enclosing characters must mirror each other. If the character has a natural mirror, e.g., ‘{’ and ‘}’ they must be used. But in a case like ‘#^’ (where no mirror of either character exists), just use the opposite order to mark the close of the variable: ‘^#’.
- You cannot use ‘<<’ and ‘>>’ as enclosing characters. They are reserved for Clause Sets, document assembly and <<*Optional/Options*>> text block use.

37.4 Multiple choice variables vs. Multichoice *Lists*

A variable can be a mere placeholder.

However, a variable can also serve the additional role of providing the possible answers for the place it is holding.

- The variable [Client's City of Residence] calls (obviously) for the location where the client resides. If there are only a limited number of possible answers, or if the answers are hard to spell, you should consider providing the choices within a 'multiple choice' variable. For example, [Newport News/Hampton/Yorktown/ Williamsburg].
- Other examples: [child/children], [he/she/it/they], [him/her/it/them].
- A multiple choice variable is presented in the Instant Database screen as a dropdown list, making it easy for the operator to select from among the choices.



A short list of multiple choices are simple to create. Just separate the choices by slashes. E.g., [chocolate/chocolate ripple/vanilla/French vanilla/Rocky Road/strawberry/ pistachio/ banana/orange sherbet (etc.)].

However, as the list gets longer and longer, it becomes more likely that a different approach should be taken. Lists like the ice cream flavors above take up too much screen 'real estate' for most applications.

Pathagoras provides a more compact way to represent long lists. We call them Multichoice *Lists*.

- A Multichoice *list* is simply a one or two word reference to what can be a very long string of multiple choices. (In our ice cream example above, the list name might be “*flavors*” and we can have dozens of them, all depicted on-screen by the word *flavors*.)
- If you place a *List* between brackets (e.g., “[*flavors*]”), Pathagoras instantly finds the various elements that comprise the list and processes it as part of the <Scan> function.

If the variable choices are short, or can be typed into the source document without creating a visual distraction because of its length, use the on-screen bracketed and 'slashed' multiple choice variables. But when the list gets long, consider Multichoice *Lists*.

Read more about [Multiple Choice Variables](#)¹¹⁸ and [Multichoice *Lists*](#)⁴⁵⁸ by clicking on the appropriate link.

37.5 <<*Options/Optional*>> text vs. variables

Optional Text is 'structural.' The chosen options are saved as part of the the final document, but are not otherwise preserved.	Variable text is 'personal' and can be saved independently from the document as well as within the final document. It can be reused.
---	--

The question frequently arises "when do I use a *variable* and when do I use an <<*Options/Optional*>> text block to hold a spot for a particular term?"

The answer can generally be determined by answering these four questions:

1. "Will the placeholder always be occupied by something?"

If the answer to this question is 'No,' then you have to be thinking "Optional" text. "Optional" text is all or nothing, take or leave it type text. Sometimes it remains, sometimes it does not.

2. "Is the content personal to the person or company for whom the document is being assembled, or is the content more structural in nature?"

When you think of 'variables,' you typically think 'personal' or 'identifying' type of information. Names, colors, values and the like are classic 'variables.' A variable is something short and unique to the customer or client, that typically could have an infinite number of responses. It is also something that you would consider saving into a permanent database record.

<*Options*> text, on the other hand, is something that is more document oriented. The choices are finite, and while saved for posterity when you save the final product to a folder, it is not saved for use (as would a client's name or address, etc) for reuse in a subsequent document.

3. "When is the most logical time to provide the 'answer' to the question? (Is the information structural or is it personal or 'finishing' in nature.)

'Options' and 'Optional' text is processed as the document initially is being assembled. If it seems that the text should be 'placed' or 'excised' before personal or 'finishing' information is provided, then think 'Options' or 'Optional'. Otherwise, think variable.

4. How long is the text.

'Options' and 'Optional' text is, compared to variables, typically much longer. That is why we call them <<*Options/Optional*>> text blocks. Variables are typically short both in name and in replacement values.

Example and discussion:

Dear [Customer Name].

Thank you for your order placed on [date of order].

Your local representative is [Name of Rep]. [He/She] will contact you within 5 business days to discuss installation.

<<*Optional*Please note. Due to the special pricing of these items, all sales are final.>>

[Name of Order Taker]

In this example, most entries are variables. Each need a 'personal' replacement value in order for the letter to complete. Each variable is short and each calls for information that is personal to an actor. Each answer is something that is appropriate to save into the customer's data record.

Sometimes all sales are final, sometimes they are not. So sometimes the last line of the letter (above the signature) will remain in the letter and sometimes it will not. Regardless, the text is 'structural' to the letter, not personal to the buyer. Plus, it is (relatively) lengthy. Therefore, it is appropriately an <<*Optional*>> text block.

A note of assurance: When choosing between using a 'variable' vs. 'options/optional' text, keep in mind that your initial choice matters little in the big picture. Don't fret it. If you choose

one and, upon reflection, the other seems the more appropriate, you can readily change it. Switching from <<*Optional/Options*>> text to [bracketed variables] is simple a matter of replacing a few characters in the body of your text. That, quite simply, is the beauty of 'plain text document assembly.'

37.6 <<*Options*>> vs. Multiple choice variables

<<*Options*>> text blocks, multiple-choice variables and *Lists* each allow you (as the author of the source text) to provide choices of text from which the user may select. The main differences are these:

- <<*Options*>> is generally used for larger text blocks. By larger, we mean anything from a sentence to several paragraphs to several pages. (However, if the text is longer than a few paragraphs, you really should consider making the text a separate clause and select it, or options to it, from the Clause Selection Screen. But, given Pathagoras' flexibility in this regard, you can do it with <<*Options*>> if that is your preference.)
- Variable are typically shorter, typically encompassing only a few words and spanning at most a line or two.
- <<*Options*>> are selected at document assembly time. (This is an important difference and concept to understand, so I will repeat-- <<*Options*>> are processed at document assembly time.) Here is how it is done:

When you press the **Next>>** button after having selected clauses from the Clause Selection Screen, Pathagoras quickly assembles those clauses into a single document. Immediately after that, Pathagoras returns to the top of the document and searches for any <<*Options*>> and <<*Optional*>> text blocks in the document. If it finds any, Pathagoras pauses to ask if the text block should be kept (<<*Optional*>>) or which option(s) should be kept (<<*Options*>>). The choices you make become part of the final assembled document. They are not 'stored' as part of a reusable data record.

- On the other hand, variables and *Lists* remain intact in the assembled document until the InstantDatabase or *GotForms?* routine is run. So, after the document is fully assembled, the variables still await completion.

37.7 <<*Options*>> vs. Clause Selection Screen

Every document in every folder (and every clause in a glossary) is technically 'optional text.'

- It is 'optional' because you can choose it or not choose it when it is presented in the Clause Selection Screen.

Sometimes, however, you don't want the user to have to select from among every single variation of clause on a particular subject matter. That could end up presenting too many options to the end user, and that can be counter productive. Sometimes (actually, frequently) it is more efficient to provide the user with some text selection options *within* the source clause. That is what <<*Options*>> and <<*Optional*>> text blocks are all about.

- If, let's say a 'shipping option' is all that distinguishes potentially separate Clauses A, B & C, it may be more efficient to combine the three clauses into one, and provide an

<<*Options*>> block which can present the mailing options.

Only actual use, and trial and error, will fully explain the difference between <<*Options*>> text and Clause Selection Screen text, but such trial and error will eventually lead you to the best way of producing documents.

But see the next section for a further discussion and comparison of these practices.

37.8 <<*Optional*>> vs. Selecting Clauses

Generally speaking, the 'best practice' approach to document assembly is 'from the ground up.' This approach emphasized throughout this manual.

- This approach is exemplified (among other examples) by the example provided in the Will book of the DemoDocs library. If you study that book (which ultimately is a simple Word folder), you will notice that it is comprised of about 30 separate documents, each document containing but a single snippet of text. Each snippet in turn represents a particular clause that can be can be use (or not) to create a much larger Last Will & Testament. So, when the user wants to create a Last Will & Testament (or a contract, or whatever), the user selects the appropriate book and the various clauses within that book are presented onto a Clause Selection Screen. The user then selects the desired clauses and the individual pieces are brought together into a complete document containing only the selected clauses.

On the other hand, it is possible to start with a single document (albeit a large one) that contains every possible Will, Contract, Bid, etc., clause for that particular topic. By placing <<*Optional*>> and <<*Options*>> markers around the appropriate optional text blocks, you can make it so that the user will make decisions as to what stays and what goes once the template has been brought to the screen. There is nothing 'wrong' with this approach but keep in mind the following:

- the more <<*Options/Optional*>> blocks you create, the greater the possibility of typographical and structural errors.
- it is more difficult to edit clauses surrounded by the administrative text required by <<*Options/Optional*>> blocks.
- it is much faster to affirmatively select well named clauses from the Clause Selection Screen than it is to process numerous individual "Do you want to keep the highlighted text" questions. (The exception to this is when you place <<*Optional*>> text throughout the document, use !GroupNames! to tie them together. This is a most proper use of <<*Optional*>> text and, because only one question is answered, it is undeniably faster than selecting individual clauses.)
- if all selectable text resides in a single document, you will not have the benefit of having a collection of individual clauses that you can insert at will.

Ask yourself this: What if you accidentally say 'No' to the question, when posed, "Do you want to keep this clause?" How will you get that clause into the document. With the 'template' approach, unfortunately you may find yourself opening the template, hunting for the clause, copying and then pasting it into the document.

Of course, this is counter-productive. If the clauses are maintained separately in a book, adding a new/missing clause is a very simple process.

Now there is a proper time and place for optional text. We just recommend that it be used judiciously, and with full awareness of the options available.

37.9 Moving Text from Source to Final Product

Pathagoras provides the user with alternative, complementary, devices for moving text from a collection of source materials into your final product. Perhaps you are building a new document from scratch or editing an existing document and want to bring in just 'one-last-term' to complete the project. Below we compare and contrast the available "text-moving" tools.

37.9.1 Clause Selection Screen vs. DropDown lists

To create a new document, you would typically click the Document Assembly icon, and then select a book. The Clause Selection screen next appears. From that screen you would select one or more of the library of clauses showing in the left panel and move them over to the right panel. Click <Next> and Pathagoras will quickly assemble the selected clauses.

If you frequently find yourself selecting only a single clause from a particular document assembly book (such as when each item in the book is a complete document or form), consider making a 'DropDown List' from that book. A DropDown List sits in the menu area of the Word screen. It is always on and always ready. When you need a single item to complete the job, click on the list, select a term, and blink-of-an-eye instantly, the term is inserted into your document.

NOTE: As you may have surmised from the above, you can maintain a Document Assembly book and a DropDown List both pointed simultaneously to the same folder. Use Document Assembly (and the resulting Clause Selection screen) when you want to select multiple clauses to quickly build the outline of your document, and if you need one or two additional fillers, use the DropDown List. The symphony you can create is incredible.



If you have a DropDown List, but do not have a corresponding 'book' in your Document Assembly library, not to worry. You can create a 'Clause Selection' screen on the fly directly from your DropDown List. (1) Scroll to the bottom of the DropDown List and click "Other Settings and Actions." (2) When the new screen appears, click the long blue bar titled 'Display as Clause Selection Screen'

See:

[Clause Selection Screen](#)  52

[DropDown Lists](#)  356

37.9.2 Clause Selection Screen vs. Clause Sets

Actually these are not different so much as the latter is a shortcut built within the former. As you use Pathagoras regularly, you will find yourself building routine documents by checking off the same clauses time after time. Instead of reinventing this wheel each time you compose a basic contract or an 'I Love You' will, you can create a Clause Set of those routine clauses. To create a clause-set, start building the document in the normal fashion (moving clauses from the left panel to the right panel of the Clause Selection Screen), but before clicking the <Next> button, click the <More> button at the bottom of the screen. Check the "Clause Sets" box and

then click <Next>. Pathagoras will create a list of the clause references and will show you how to save that list for reuse as a clause-set.

The next time you display the Clause Selection Screen, the clause-set you created will appear in the list of available terms along side of the other terms. Now you only have to call in one item, not the several (potentially dozens of) terms the clause-set represents.

37.9.3 Mouseless vs. everything else

As you work with your clauses more and more, you will begin to memorize their names. When that becomes the case, you may decide that it is simply easier to type the clause name onto your screen and retrieve the clause with an <Alt-G> keystroke. Very little could be simpler or more elegant. See the Mouseless Operations (<Alt-G>) Manual for more information and for a wide range of examples and uses of <Alt-G> for Pathagoras operations. And see the below section regarding naming techniques for the most efficient set-up of your system.

If you have adopted the prefix/suffix style of naming your specialty documents and clauses, you will find mouseless operations to be an incredible time saver.

37.10 Personalizing Your Documents

The previous discussions focused on the alternative methods for inserting variables, lists and alternative text blocks into your source documents.

We will now turn our attention to the 'last' part of the process: *replacing* variables in the now almost final document in order to personalize the document for the specific client or customer. Pathagoras provides several complementary ways to replace variable text with personal data.

37.10.1 GotForms? vs. InstantDatabase

Pathagoras provides two distinct ways of replacing variables in your final document with personal data. The first is GotForms? and the other is InstantDatabase.

1. **GotForms?** is a one-at-a-time replacement tool. It only works when variables appear between brackets or as underlines. Multiple choice variables and *lists* appear as selectable buttons on the **GotForms?** screen.
2. InstantDatabase is an all at one time variable replacement tool. Multiple choice variables and *lists* appear as dropdown lists within the IDB screen.

37.10.2 InstantDatabase Mask vs. InstantDatabase Scan

One of Pathagoras' most impressive features is its ability to instantly scan a document for bracketed variables and to return those variables onto the left ('variables') side of the InstantDatabase screen. From there you can supply the personal information, and if desired, save the pairings of variable with personal data for reuse with a similarly variable-coded document.

But what happens when the information called by the variable is not self-evident either in terms or content or format? Creating a mask is frequently the 'best choice.'

Advantages of a Mask:

- You can set the order in which the variable elements are completed. When you Scan

for bracketed variables, they appear on the screen in the order found in the document, which is not necessarily the most convenient completion order. (Perhaps you have an intake sheet and would like to complete the elements in the order in which they appear on the intake sheet.)

- You can complete all potentially usable information regarding a client or customer in one sitting. A mask (hopefully) will contain the variables needed for all documents that the client/customer may require. Ditto the intake sheet remark from the previous paragraph.
- You can categorize your data with a mask. At the top of each ‘page’ of the mask is a text box that you can use for this purpose. Categories make it easier for the end user to understand the purpose of each variable.
- You can assign variable completion ‘tips’ and ‘examples’ to a mask. E.g., if the variable is [Date of Completion] what does that mean in terms of format? Is it “4/27/06” or perhaps April 27, 2006. With a mask you can explain it, and provide examples.
- You can attach a mask to a document assembly book. When the assembly routine is complete, and you call up the InstantDatabase to replace variables with personal data, the mask will automatically appear.
- You can add variables to any document using the variables stored with the mask. Click the Pathagoras dropdown menu and select variables. This makes neutering documents for your document assembly system faster and more consistent.

Advantages of Scan:

- A scan is quick and requires absolutely nothing in the way of setup. It is <Scan> and you are done. (Of course, the source document has to have bracketed variables within it in order for a <Scan> to return results.)
- Aside from its simplicity, scanning has few advantages over taking the time to create a functional mask. But then, there is something to be said for simplicity. If you are happy with scanning, stick with it until you perceive that the benefit of a mask outweighs the speed and simplicity of the scan.

37.11 Storing Your Source Documents & Clauses

37.11.1 Folders vs. Glossaries

Okay. This is a big one. In what manner, and where, should you store your content? Here we mean your ‘source’ documents and clauses, those from which future documents will be built.

Pathagoras calls the locations where source documents are stored “books.” Books (up to 10 at a time) are collected into libraries. A book can be either:

- a folder of documents (each document perhaps containing a single paragraph of text that is a building block to a final product) or
- a glossary (a single document that contains all of the building block clauses, each segregated from the other by bookmarks).

Pathagoras doesn't care if a book is a 'folder of documents' or a 'glossary'. It handles both methods well and in essentially the same fashion. Any speed difference is generally negligible.

The choice is personal to the user.

That said, let's go head-to-head.

Why use folders:


- They are easier to conceive and initially easier to work with.
- Everybody knows what a document in a folder is.
- What you have already on your computer is immediately available for assignment to a Document Assembly book. To assign a folder to a book, all you need to do is to navigate to the folder one time. After that, the book is set.
- To edit a clause in a folder, you simply have to locate the file, edit the text and then save the file. Done.
- Add more documents to the folder, and Pathagoras will automatically add that file to the Clause Selection Screen list the next time you begin document assembly. Little could be easier.
- Glossaries are a conceptual 'step-up' for the more advanced user. For newcomers to the program, we recommend use of folders.


Why use a glossary:

- Portability: Glossaries are more portable. Since all clauses are contained in a single document, if you want to transfer your entire 'book,' you simply copy one file.
- Editability: Glossaries are easier to edit. Since all clauses are contained in a single file, if you wanted to change, let's say, a variable (or any other text) that appears in a glossary, you can perform a Find and Replace (Word function) and replace every instance of the old with the new text. (If each term is in a separate document, you would have to open every document to make the same type of change.)

Making the decision:

- So, with that explanation, which one should you choose. Well, it really doesn't matter. Use the one you think you will feel most comfortable with.
- It is easy to start with one and later move to the other. And if you decide to return to the original style, switch back with no problems.
- Our most important point here is '**don't worry about it.**'
- Pathagoras contains the easy to use conversion tools to make this happen.

➤ See this page to convert a [glossary to a folder](#) .

➤ See this page to convert a [folder to a glossary](#) .

37.11.2 Documents vs. Templates

This section isn't so much a description of alternatives in Pathagoras as it is a discussion of alternatives to Pathagoras.

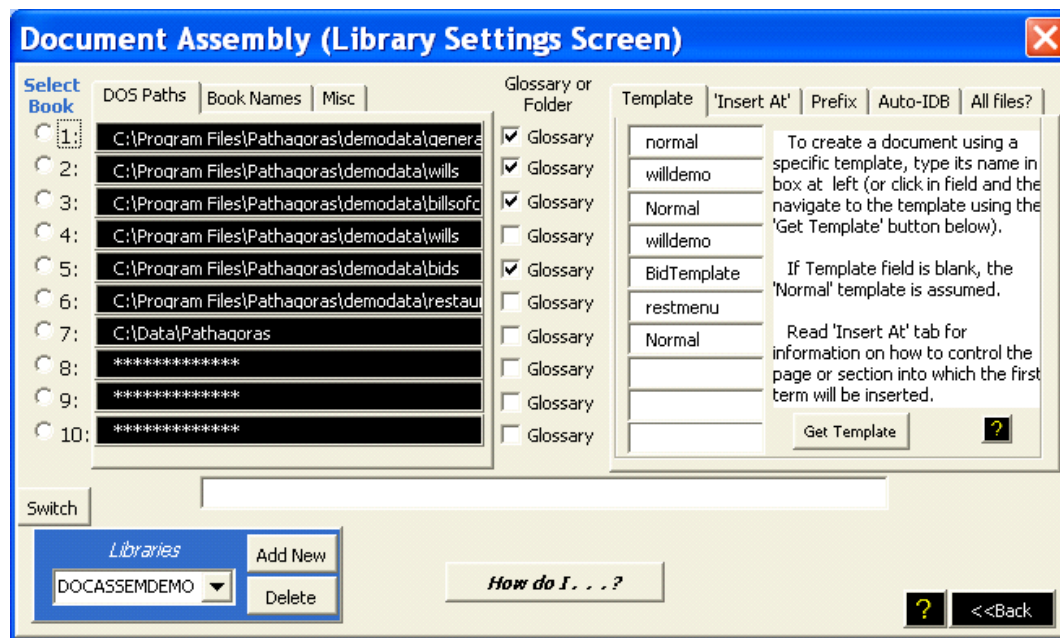
Almost every major document assembly program out there encourages you to create

'templates' as your source document. Pathagoras says create documents for your source documents. As used in Pathagoras and as contemplated by Microsoft, a template gives shape to a document, but is not the document itself. Templates are typically blank or contain only a few lines of introductory text. They are used to give shape (margins, styles and fonts, headers and footers, etc.) not content. (Content absolutely is allowed, but it should be minimal.)

Templates are the preferred source for many programs because templates don't normally open as themselves, but open as copies of themselves (so you don't overwrite them). Pathagoras handles that by only opening copies of the original during the document assembly process, not the original. See the Guide to Templates for more information and examples.

The 'why?' for Pathagoras' preference for source clauses being built on a document as opposed to a template platform is simple. Have you ever saved a template? Word forces a template into a template directory. It requires you to navigate back to the folder into which you really want to save the document. Documents are simply easier to save, easier to work with and are the default 'search' type of file that Word uses when displaying its File Open dialog. The reasons are not earth shattering, but neither are they insignificant. It is just easier to work with documents than with templates.

That said, templates are important in the proper use of Pathagoras. When used to assign shape (margins, styles and fonts, headers and footers, etc.) to the document that will be built on top of it, template cannot be underestimated. If you have a particular 'style' template that you wish to associate with a document assembly book, create the association in the Document Assembly <Settings> screen. Type the templates name (or navigate to it using the Get Template button).



Because of their importance, Pathagoras dedicates a whole screen to the subject of templates, but even there, the emphasis is aimed at the 'shape-building' template, not the complete document 'template.'

The Templates screen can be found in the Pathagoras features dropdown list.

See [Templates](#) ⁹⁶

37.11.2.1 Mouseless vs. Everything Else 2

There is simply nothing easier than typing a number representing a known folder and pressing <Alt-G>. Try it. [Click here](#) ⁵⁵⁰ for more info. Need a chart to help you with the numbers -- [click here](#) ⁵³⁹.

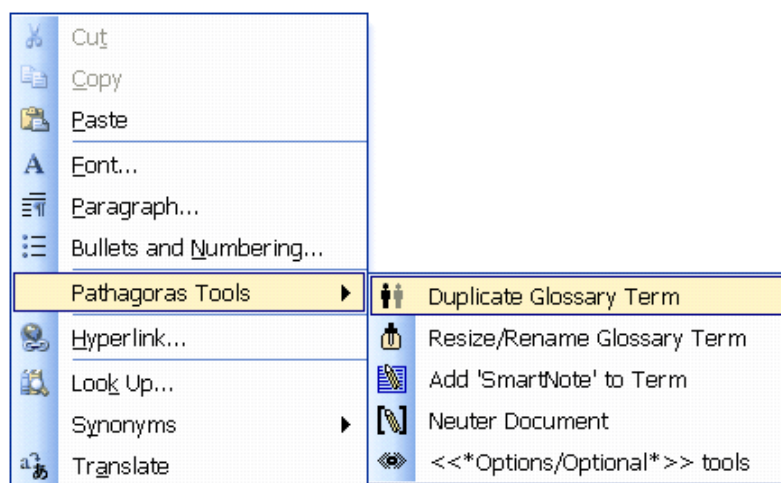
37.11.2.2 Using PathSmart Screen vs. Word 2

You can quite literally navigate your entire system without any 'actual' (i.e., clicking up, down and across the folder trees) navigation using the [PathSmart](#) ⁵³⁸ screen. You can remain on the same screen while searching for a particular file. Display of a filtered list occurs much faster via PathSmart than through Word alone.

37.11.2.3 Direct Add vs. Duplicate 2

Sometimes you are in the glossary because you want to copy one of the terms, modify it slightly and give it a slightly different name to reflect that difference. You can highlight, copy and paste the text you want to duplicate, and then perform the 'highlight & add' steps to accomplish this. An easier way makes Pathagoras do the work.

1. Right click inside of the term.
2. Select 'Pathagoras Tools' from the right-click menu
3. Select 'Duplicate Glossary Term.'



As the name of the command suggests, Pathagoras will duplicate the term. It will also let you give the term a new name (proposing the original name with a letter designator) and a new subject.

4. Once duplicated, you should modify the clause to reflect its differences from the original clause.

37.11.2.4 'Highlight & Add' vs. Direct Add 2

Pathagoras offers a number of ways of capturing text and adding it to your books. These two are your most likely options:

- **Highlight & Add** requires only that you (1) highlight the text you want to add and (2) press <Alt-G> next to the highlighted text. The Term Works! screen will appear where you can name the clause, provide its subject and select the glossary into which the term should be added. You do not need to open the glossary—Pathagoras handles that for you (as well as placing the term in alphabetical order according to its name).
- **'Direct add'** involves opening the glossary and either pasting text from another source or typing original text. Then highlight the text and press <Alt-G> (i.e., highlight & add). As you can see, the method at the first bullet probably makes more sense unless you happen already to be in the glossary for other purposes – see below.

37.11.2.5 Bulk Add vs. Instant Book 2

Pathagoras can easily handle 'mass transfers' of documents into a glossary. The two primary methods include what we call Bulk Add and Instant Book. Each presupposes that you have a complete document (preferably one with all of the bells and whistles might be contained in a final document) that you want to disassemble into component pieces and assign names to them. From those disassembled pieces, you would duplicate some or many of them. Your objective here is to provide variety to the selection of clauses so that when you decide to assemble a document, you have more pieces from which to choose that were present in the original work. Again, the specifics are discussed in other manuals. The head-to-head comparison goes like this:

Bulk Add: You segregate each clause using any set of enclosing markers. "<&" to begin and "&>" to end works fine. Just inside of the opening marker, you assign a name to the clause and a subject name. Separate name and subject by a simple 'slash.' The subject and the clause text likewise are separated by a slash. Since you control everything from the keyboard, you have great control over all content. You do not have to adopt the prefix/suffix naming convention (although you do have to follow bookmark naming rules if you are going to create a glossary, and document naming rules if you are going to create a folder of clauses).

Instant Book: Easier at first and automatic. All you do is place a unique marker (anything will do, but we suggest "(*)" as an appropriate marker) at the beginning of each term that you want to become a separate term. Click Instant Book and Pathagoras takes over. Benefits: ease. Detriments: Pathagoras will insist that the names follow the prefix/suffix naming convention. That is not a bad thing unless you don't want it. If that is the case, consider Bulk Add.

37.11.2.6 SuperBooks vs. other locations 2


For the most efficient document management (and for more efficient document assembly that results from best practices in document management), you should store your source clauses and forms in well organized folders and sub-folders that make retrieving your clauses and documents as logical as possible.

But some documents and clauses don't fall into a particular category. They are too general in nature.

- Standard cover letters, non-subject specific correspondence, etc. cannot be readily categorized by topic (other than "General.")

- Addresses and phone numbers don't neatly fall into 'subject areas' either. These are, however, items to which you may wish instant access.
- Such standard forms can be lumped together under a folder perhaps called "General Office Forms."

Items that cannot be readily categorized, but for which you want instant access should be stored in a "SuperBook." If Pathagoras cannot find the term you seek in a subject specific location, it will look into the SuperBook to see if it is there.

 You do not have to use the prefix/suffix naming conventions in order for Pathagoras to look in the "Super" locations for the requested document. Assuming that there is not a better place for the document or term to be saved, store the item in one of the SuperBooks.

See [SuperBooks](#)⁹⁰ for the details.


37.11.3 SuperBooks vs. other locations

For the most efficient document management (and for more efficient document assembly), you should store your source clauses and forms in well organized folders and sub-folders. That should make retrieving your clauses and documents as logical as possible. We also recommend that you name your clauses using the prefix/suffix naming style. That way you can instantly recall a clause simply by typing its name and pressing Alt-G beside it.

But some documents and clauses don't fall into a particular category. They are too general in nature.

- Standard cover letters, non-subject specific correspondence, etc. cannot be readily categorized by topic (other than "General.")
- Addresses and phone numbers don't neatly fall into 'subject areas' either. These are, however, items to which you may wish instant access.
- Despite their general nature, you still want to be able to have instant "<Alt-G>" (mouseless recall) access to these kinds of document assembly elements.

For such a need, consider lumping those items together under a single folder. Perhaps call it "General Office Forms." And then designate that folder as your "SuperBook." That way, if Pathagoras cannot find the term you seek via mouseless recall in a subject specific location (as indicated by a prefix), it will look into the SuperBook to see if it is there.


 You do not have to use the prefix/suffix naming conventions in order for Pathagoras to look in the 'Super' locations for the requested document. Assuming that there is not a better place for the document or term to be saved, store the item in one of the SuperBooks.

NOTE: Some of the mouseless assembly 'issues' can be resolved by judicious use of the 'SuperFolder' and 'SuperGlossary' tools. Some users may be tempted to store all documents inside a SuperFolder. Resist that temptation. The 'SuperFolder' and the 'SuperGlossary' are intended to store office-level standard text, not subject specific text.

See [SuperBooks](#)⁹⁰ for the details.

37.11.4 Local vs. Network Storage

This is simply a personal or office level choice. If the documents are shared by more than one user, they probably should be stored on a network server. If the documents are personal to the user, and will likely never be used by others, then storing them locally certainly makes sense.

If document assembly speed is an issue, and the network server is in a remote location that is accessed over the Internet (as opposed to a local area network), you may wish to consider 'synchronizing' the network files to the local hard drive on a regular basis. Pathagoras has tools for this. See [synching](#) .

37.12 Adding Text to Source Books

There is a multitude of ways of adding text to your source folders and glossaries. We compare the major ones below.

37.12.1 Bulk Add vs. Instant Book

Pathagoras can easily handle 'mass transfers' of documents into a glossary. The two primary methods include what we call Bulk Add and Instant Book. Each presupposes that you have a complete document (preferably one with all of the bells and whistles might be contained in a final document) that you want to disassemble into component pieces and assign names to them. From those disassembled pieces, you would duplicate some or many of them. Your objective here is to provide variety to the selection of clauses so that when you decide to assemble a document, you have more pieces from which to choose that were present in the original work. Again, the specifics are discussed in other manuals. The head-to-head comparison goes like this:

Bulk Add: You segregate each clause using any set of enclosing markers. "<&" to begin and "&>" to end works fine. Just inside of the opening marker, you assign a name to the clause and a subject name. Separate name and subject by a simple 'slash.' The subject and the clause text likewise are separated by a slash. Since you control everything from the keyboard, you have great control over all content. You do not have to adopt the prefix/suffix naming convention (although you do have to follow bookmark naming rules if you are going to create a glossary, and document naming rules if you are going to create a folder of clauses).

Instant Book: Easier at first and automatic. All you do is place a unique marker (anything will do, but we suggest "(*)" as an appropriate marker) at the beginning of each term that you want to become a separate term. Click Instant Book and Pathagoras takes over. Benefits: ease. Detriments: Pathagoras will insist that the names follow the prefix/suffix naming convention. That is not a bad thing unless you don't want it. If that is the case, consider Bulk Add.

37.12.2 'Highlight & Add' vs. Direct Add

Pathagoras offers a number of ways of capturing text and adding it to your books. These two are your most likely options:

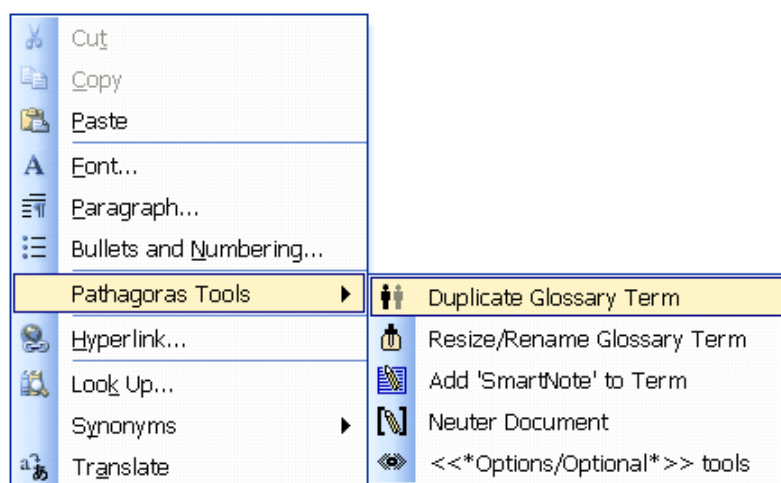
- **Highlight & Add** requires only that you (1) highlight the text you want to add and (2) press <Alt-G> next to the highlighted text. The Term Works! screen will appear where you can name the clause, provide its subject and select the glossary into which the term should be added. You do not need to open the glossary—Pathagoras handles that for you (as well as placing the term in alphabetical order according to its name).

- **‘Direct add’** involves opening the glossary and either pasting text from another source or typing original text. Then highlight the text and press <Alt-G> (i.e., highlight & add). As you can see, the method at the first bullet probably makes more sense unless you happen already to be in the glossary for other purposes – see below.

37.12.3 Direct Add vs. Duplicate

Sometimes you are in the glossary because you want to copy one of the terms, modify it slightly and give it a slightly different name to reflect that difference. You can highlight, copy and paste the text you want to duplicate, and then perform the ‘highlight & add’ steps to accomplish this. An easier way makes Pathagoras do the work.

1. Right click inside of the term.
2. Select ‘Pathagoras Tools’ from the right-click menu
3. Select ‘Duplicate Glossary Term.’



As the name of the command suggests, Pathagoras will duplicate the term. It will also let you give the term a new name (proposing the original name with a letter designator) and a new subject.

4. Once duplicated, you should modify the clause to reflect its differences from the original clause.

37.13 Navigating

37.13.1 Using PathSmart Screen vs. Word

You can quite literally navigate your entire system without any 'actual' (i.e., clicking up, down and across the folder trees) navigation using the [PathSmart](#)⁵³⁸ screen. You can remain on the same screen while searching for a particular file. Display of a filtered list occurs much faster via PathSmart than through Word alone.

37.13.2 Mouseless vs. Everything Else

There is simply nothing easier than typing a number and pressing <Alt-G> to navigate to any folder on your system. Try it. [Click here](#)⁵⁵⁰ for more info. Need a chart to help you to remember which number has been associated with which folder? [Click here](#)⁵³⁹.

37.14 Word Equivalents

Many functions in Pathagoras have parallel functionality in Word. We call these 'functional equivalents'. The difference is that Pathagoras allows plain text, whereas with Word, you must know how to create a field.

Word Field or Function	Pathagoras Equivalent	Comment
Today field	[Today]	Once replaced, the [Today] date becomes static. In Word, it changes each time you open the document on a new day, unless setting changed.
Any variable	[bracketed term]	
AutoText	any word followed by Alt-G	Autotext is limited to user who created Autotext. (Entries are stored in users 'normal.docx' file. No such limit exists for Alt-G.
IncludeText	<<any document>>	If document name is in HuntPath, no folder designator is required.

The Pathagoras System

**Adobe™ Acrobat™
Interactivity**

Part

XXV

38 Adobe™ Acrobat™ Interactivity

Beginning with version 2015, Pathagoras has substantial Adobe™ Acrobat™ interactivity.

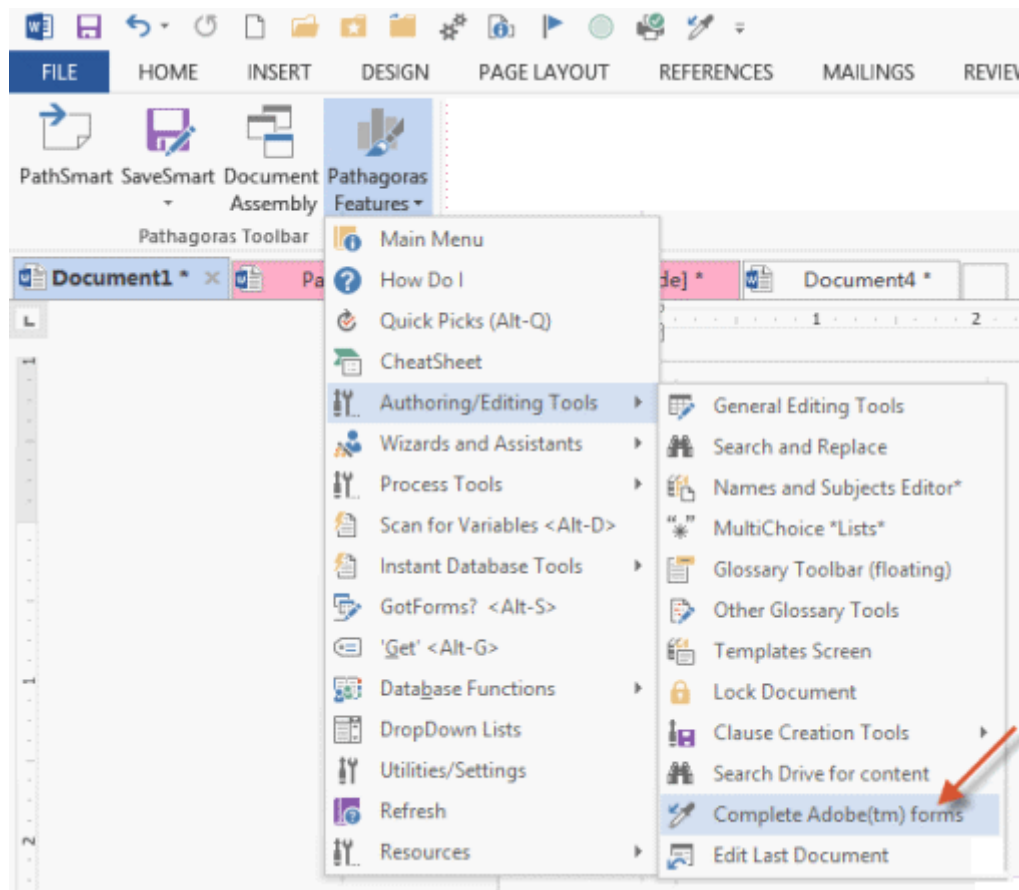
1. Pathagoras can complete Adobe Acrobat forms as if they were Word documents. For example:

• probate forms	• court filing cover sheets	• tax forms
• uncontested pleadings/ motions	• reports of adoptions	• immigration forms

These forms are regularly published by federal, state and local courts in Adobe Acrobat. More forms are coming daily. Many of these forms have been made mandatory. You may not have a choice about having to use them, but you now have a choice about how you can complete them

2. With Pathagoras, you can complete Acrobat PDFs as easily as you can Word based documents. (PDFs will never replace specially crafted contracts, agreements, complex pleadings, wills & trusts, etc. So there will always be a need for true document assembly. But now one program – Pathagoras -- can handle the full range of forms used in a typical office, from the ‘fill-in the blank’ Acrobat PDFs to the most lengthy and complex of documents.
3. You can work with Acrobat PDFs in several ways, summarize below and explained with more detail in the following pages:
 - a. Pathagoras can complete PDFs into which you have inserted [bracketed plain-text variables]. [Click here to read more.](#) ⁶⁴⁵
 - b. Pathagoras can scan an Acrobat form for either [bracketed variables] or the 'raw' field names of an Acrobat form. It can those names as variables onto Pathagoras' familiar Instant Database form. [Read more here.](#) ⁶⁴⁷
 - c. Pathagoras lets you assemble individual PDFs into a single PDF. This feature is similar to the document assembly performed by Pathagoras in its standard Word operations. [Read about PDF assembly here.](#) ⁶⁵¹
 - d. Pathagoras can scrape the usable data from completed Acrobat PDFs and pull the data directly into your Pathagoras system. [Read more here.](#) ⁶⁴⁸

Call the Acrobat module from Pathagoras Features | Authoring Editing Tools. Click the 'Adobe' element as shown below



If you anticipate using this feature a lot, you should place the Acrobat 'button' onto your Quick Access Toolbar (the 'QAT'). Just right click on the 'Complete Adobe forms' and select the QAT option. That will provide instant access to this feature, and save you many clicks.

Please note:

1. The interconnectedness discussed above is with Pathagoras and the Acrobat™ brand. At present Pathagoras cannot complete other PDFs. This may soon change, but for now be aware of the restriction.
2. There are two basic types of PDFs:
 - a. Letters and other documents that have been converted from a native format such as Word into a PDF document. That converted document is essentially an image of the original, and is more difficult to modify.
 - b. PDF forms with 'fields' that are intended to be completed by the end user. The fields typically can be highlighted by clicking the 'Highlight Form Fields.. Filling and scraping can be done only with PDFs of this type. Information in a type 'a' letter is not scrape-able because all of the text is essentially the same.

38.1 Replacing using [IDB] variables

Pathagoras can complete PDF fields which currently reflect [bracketed plain-text variables] (i.e., the PDF has been 'Pathagorized' in much the same way that you would 'Pathagorize' a Word document).

To 'Pathagorizing' a PDF, simply type a bracketed variable in each location you want to replace with data from your Instant Database source. For example:

LASRIS, WALSH & ASSOCIATES, P.C.
Confidential Background Information
Domestic Relations (Divorce or Separation)

Date: _____, 20____ ☐ Divorce ☐ Separation ☐ Advice

Please Answer all Questions Completely

All information which you provide to us will be held in strictest confidence. If you feel uncomfortable about answering any question, please feel free to omit the answer. If you feel you have been provided an incorrect form, please let the receptionist know.

1. Your Full Legal Name: [Client Name] (Jr./Sr./____)

Address: [Client Address] City: [Client City] State: [Client ST] Zip: [Client Zip]

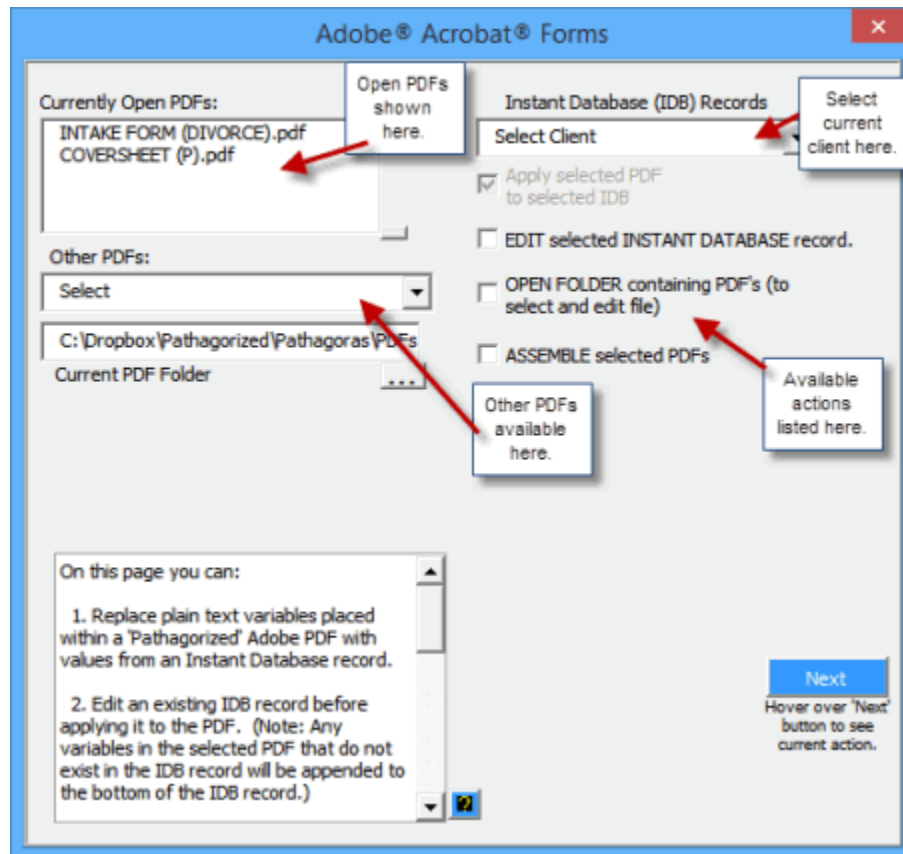
Email address: [Client Email] Cell Phone: (____) [Client Cell Phone]

Work Phone: (____) [Client Work Phone] Home Phone: (____) [Client Home Phone]

Save the now Pathagorized document into your collection of PDFs.

When you are ready to 'complete' the form, either call up the PDF first (in which case it will appear in the 'Currently Open' list when you activate the PDF module, or simply call up the PDF module. It simply is "Pathagoras Features | Authoring/Editing Tools | Complete Adobe Forms". (The previous screen illustrates the steps.)

The following screen will display:



Select the PDF you want to edit/complete, either from the "Open PDFs" list box or from the dropdown list labeled "Other PDFs"). Then select the client or customer's record from the dropdown list at the right (labeled "Instant Database (IDB) Records". Then click the Next button. Pathagoras will instantly and seamlessly call up the selected PDF and apply the selected IDB record to it.

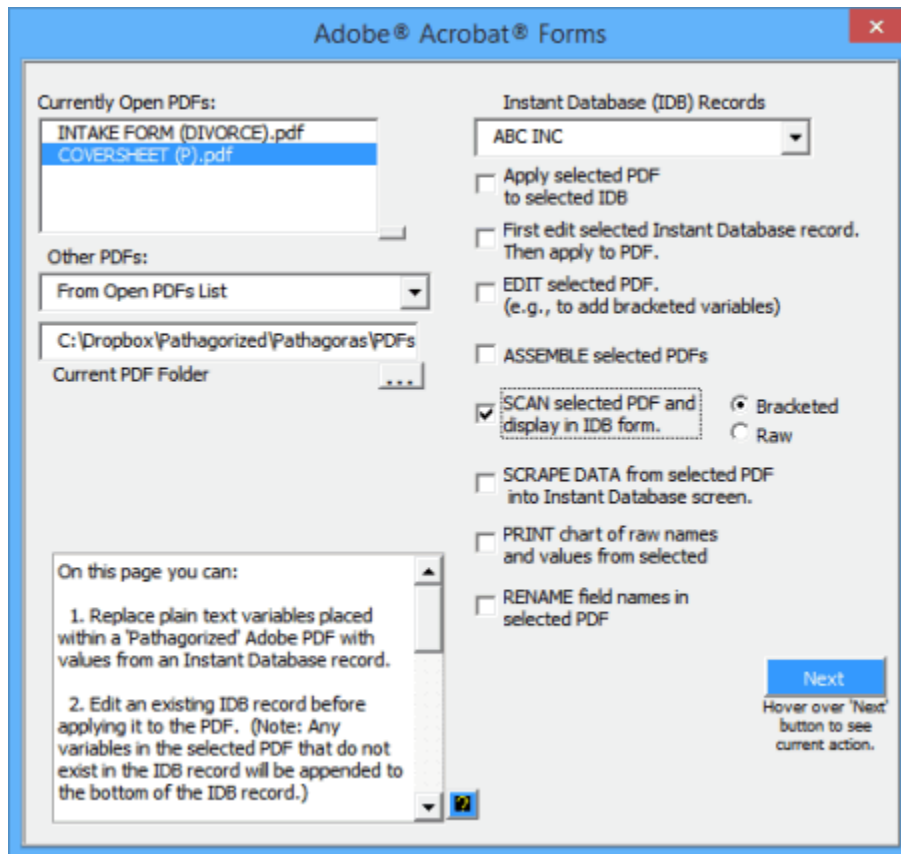
NOTE: If you need to edit the IDB record first, either to check/correct data, or to add fields in the PDF that may not be in the IDB record, click the second box from the top. Then click the Next button. Pathagoras will display the data on the IDB screen for appropriate editing. When done, click the Next button on the IDB screen and Pathagoras will fill out the PDF.

In a matter of seconds, you have a completed form using the data contained in your Instant Database system.

38.2 Adobe and Scanning

Pathagoras can scan an Acrobat form for the 'raw' names of the text boxes. Pathagoras will identify display those names as variables onto the familiar Instant Database form.

All you need to do is assign personal values to replace the variables. To set this up, select a PDF that you want to scan. If you want Pathagoras to scan the PDFs variables into an existing Record, select that Record. Check the Scan box from the Action column.



Press the Next button on this screen and Pathagoras will open an IDB screen and place the existing variables and their values onto the screen. Pathagoras will immediately scan the PDF for any 'new' variables and append them to the bottom of the IDB screen. Complete or edit the variables as needed.

Press the "Next" button on the IDB screen. Pathagoras will save your IDB changes and then instantly fill in the fields in the PDF.

The personal values can be saved (just like with 'regular' Pathagorized forms) and can be reused with other Acrobat forms AND with other Pathagorized Word documents.)

38.3 Adobe Data Scraping

Pathagoras can 'scrape' the usable data from completed Acrobat PDFs. Let's say you have send an Acrobat form to a client or customer to complete. When it is returned, you can tell Pathagoras to pull the data from the form and import it directly into your Pathagoras system.

To scrape data from an Adobe Acrobat form, select the form and click the 'Scrape Data' box in the Action column.

When you 'scrape' a document for its values, two elements of each PDF field are involved:

1. the *field name*. The field name is not visible when you look at a PDF form. It sits in the background, but is essential to the effective implementation of scraping. The field name serves as the 'variable name' for purposes of creating an Instant Database record.
2. the *field value*. This is the part of the completed for that you can see (and that you or

your client/customer completed.)

When the document is scraped, Pathagoras reads the invisible field names and pairs them with what is typed in the corresponding field value.

With no further setup, you can scrape any data from a PDF. The current 'raw' field names and their corresponding values will be displayed. the problem is the raw field names are probably useless. They typically have names like "Field1" and "Field2" (useless terms) or, if some thought was put into the namings scheme by the author, something like "User.Name.First" or "FirstName". While not useless, they probably don't match the names you already use in your Instant Database system. And after all the whole point of this feature is to give you usable and re-usable information. Not to worry. We have that all figured out.

So, the prerequisite to successful scraping of data that is immediately usable in your Instant Database system is that the PDFs field names must match the variable names you already use. Pathagoras makes this easy to do. Pathagoras can rename your field names automatically.

- The easiest way to change the base field names to [bracketed variable] names is to open the PDF and just type in the field the name of the variable you want the field to represent. Something like this will work:

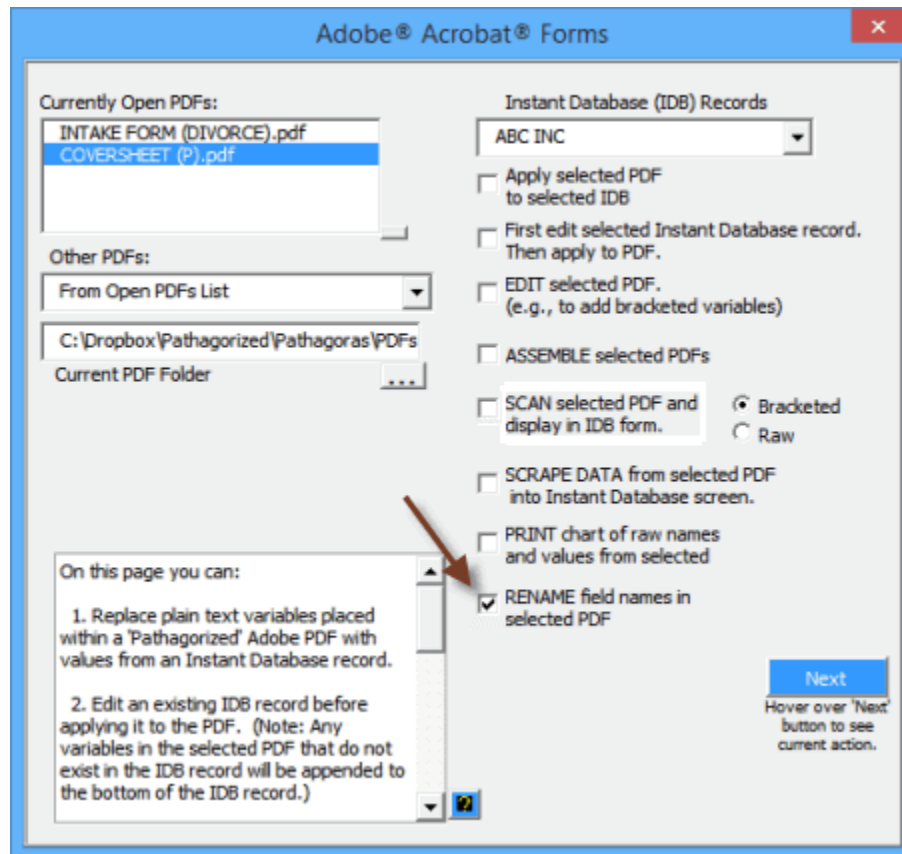
LASRIS, WALSH & ASSOCIATES, P.C.
Confidential Background Information
Domestic Relations (Divorce or Separation)

Date: _____, 20____ Divorce ☐ Separation ☐ Advice ☐
Please Answer all Questions Completely

All information which you provide to us will be held in strictest confidence. If you feel uncomfortable about answering any question, please feel free to omit the answer. If you feel you have been provided an incorrect form, please let the receptionist know.

1. Your Full Legal Name: [Client Name] (Jr./Sr./)
Address: [Client Address] City: [Client City] State: [Client ST] Zip: [Client Zip]
Email address: [Client Email] Cell Phone: () [Client Cell Phone]
Work Phone: () [Client Work Phone] Home Phone: () [Client Home Phone]

Click the Rename button on the Acrobat screen and click the Next button:

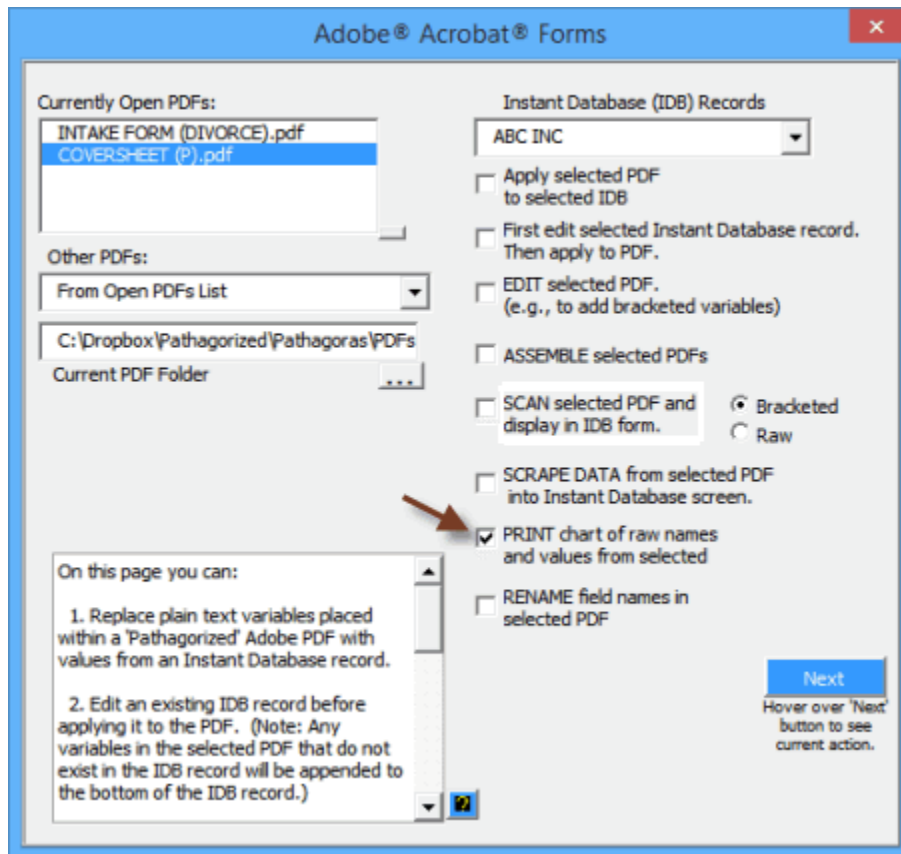


You will be presented two options. Select the first one. (It reads 'Replace the name with the [bracketed value] of each field.') The replacements take place almost immediately. Save the PDF. You can now distribute the PDF to your clients/customers for completion. When the completed forms are returned to you, you will be able to scrape them for the data they contain directly into an Instant Database record.

VALIDATED DATE FIELDS: You will not be able to place a [bracketed variable] into a field that Acrobat has been told to 'validates'. For example, if the field calls for a date, and the field is 'validated' to make sure that what you inserted was a real date, Acrobat will reject [Date of Birth]. In such case, you should put in a valid, but 'coded' date. Remember your code. (E.g., 2/22/2222 might in your 'code' stand for Date of Marriage. When Pathagoras encounters a date field that is outside of the last 100 years, it will stop and ask for a bracketed variable.)

And alternative way to rename date fields (or any field for that matter) is to manually edit the form. Instructions on how to manually edit PDF are provided by Adobe [at this link](#). Third parties provide good information on editing Acrobat forms. [Here is one example](#).

- The alternative way involves using the Instant Database screen. The left side of the IDB screen will contain the current field names and the right side will reflect the new names. To complete the left side, you can tell Pathagoras to SCAN the selected PDF, and choose the 'Raw values' options. Pathagoras provides a tool called 'Print Chart' which can list all of the field names and the values currently typed into those fields is displayed onto a table. This will help you to identify which field names you need to change.



38.4 Adobe PDF 'Assembly'

This feature is similar to the document assembly performed by Pathagoras in its standard Word operations.

To assemble Adobe Acrobat forms, select the Assemble button from the Action column. The screen will expand and more instructions will be presented. Select from the PDFs dropdown list at the left the ones you want to assemble. Pathagoras will place your selection in the 'To be assembled' panel at the right. You can rearrange your choices, and delete any erroneously selected choices. When ready to proceed, click 'Assemble'.

The Pathagoras System

Miscellaneous Matters

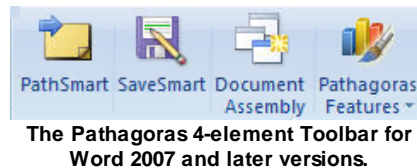
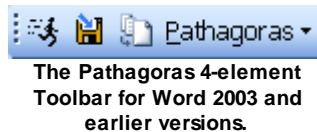
Part



39 Miscellaneous Matters

39.1 Four-element toolbar

When you install Pathagoras, a four element toolbar is added to the toolbar:



The elements are these, from left to right:

- **PathSmart:** The Quick Find aspect of Pathagoras' document management system.
- **SaveSmart:** The Quick Save aspect of Pathagoras' document management system.
- **Document Assembly:** Quickly assemble new documents from existing component parts.
- **Pathagoras drop down features list:** everything else is displayed when you click on "Pathagoras".

See also:

[Two element toolbar](#)  655

39.2 Repointing after moving files

As you build more and more libraries, and populate them with books, you may wish to better organize things at the Windows level. Perhaps you want to move your estate planning documents to a folder that is separate from your Real Estate clauses. Perhaps you want separate your Domestic Relations forms into separate 'Pleadings' and 'Agreements' folders. Perhaps you want to move files to your server, or into the cloud.

Pathagoras encourages this kind of reorganization. To accomplish the moves, just use your normal Word/Windows tools. But when you finish, you need to make sure that the pointers you set in Pathagoras are still functional.

If you move a book, you do need to make sure that the pointers reset to the new locations. Pathagoras has no way to know where you moved a particular book, so repointing must be performed manually.

There are 5 pointers about which you need to be concerned when you move files. Not all will be relevant to your situation, but we list them all here. They constitute the common 'culprits' to look at if things are not working correctly. All are very easy to correct.

1. **Document Assembly Settings:** If you have moved one or more folders or glossaries containing your source text, re-point any books which referenced that text by following the steps [outlined here](#)⁷⁸.
2. **Instant Database Files:** If you want to sharing data among all users, you need to move the data records to a common location. Once you have done so, follow the steps [outlined here](#)²⁵⁰ to point your system to that spot.
3. **Prefix pointer settings:** Pathagoras maintains a table pairing 'prefixes' with the folder or glossary assigned to that prefix. (Read more about the [prefix/suffix naming convention](#)⁴²⁰ here.) You can manually reset a prefix pointer via the 'Utilities/Settings | All Settings | Prefixes' screen. But let's assume that, following the steps in paragraph 1 above, you have reset your pointers to the various books which coincide with prefixes. A faster way to reset the prefix pointers may be to Display the Settings. Then, click the tab marked Prefix and click the Assign button. This will re-point all prefixes listed in the Library.
4. **SuperBook:** If the book you have moved is one of your SuperBooks ('SuperFolder' or 'SuperGlossary'), make sure that you re-point the SuperBook's setting to the new location. Display Utilities/Settings and select File Locations tab. Click the black 'Set' button to the left of the box labeled SuperFolder or SuperGlossary. Follow the prompts to re-point the link.
5. **DropDown Lists:** If the folder or glossary is also reflected in a DropDown List, be sure to perform a «Replace» operation. «Replace» is found in the Other Setting and Actions screen found at the bottom of the DropDown List.

39.3 Two-element Toolbar

Some users do not plan to use the document management features of Pathagoras. They have installed and fully implemented other such systems and do not wish the 'PathSmart' and 'SaveSmart' elements to appear on the toolbar.

To show just the Document Assembly icon and the Pathagoras Features menu, display the Utilities/Settings screen. In the "At a Glance" section on the front page, click the "Miscellaneous" tab. Check the "Show 2-button Pathagoras toolbar" item in the 'Specialty Settings' section.

39.4 Navigation via Textboxes

Whenever a Settings or other Pathagoras screen displays a text box containing a DOS path entry, you can open the referenced folder by double clicking inside of the text box.

- For example, the File Locations tab under Utilities/Settings contains fully qualified DOS path addresses. Double click inside one and Pathagoras will open the folder with a standard File Open dialog.

39.5 Other Navigation Tricks



Need to open the folder of the currently active document?

- Click the SaveSmart icon in the toolbar.
- Click the unlabeled button in the lower right side of the SaveSmart screen. (When you hover over the button, it will display "Open the folder that contains the underlying document.")
- The folder of the current document will be displayed.

39.6 Pointers (in general)

90% of Pathagoras' functionality relies upon a very simple principle: Pathagoras accesses the various source documents and data files not because it creates new material, but because it simply points to files and locations already present on your computer.

Please click on [this link](#)²¹ to read more about this pointer concept, and its powerful implications.

39.7 Default Settings

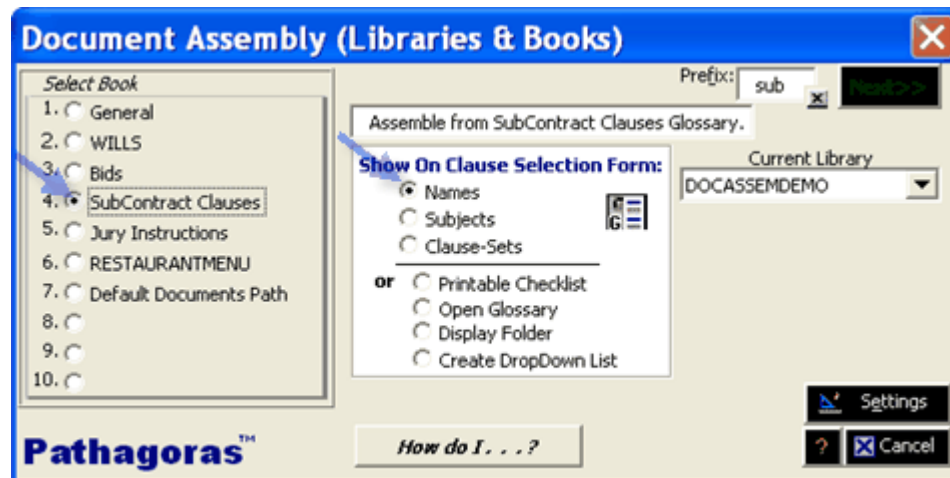
You can create or change 'default' settings on any 'action' screen by pressing the Shift button before clicking on the item.

- A 'default' setting is one that will be pre-selected when you first enter or call up the screen.
- 'Action' screens are those screens that lead directly to a primary Pathagoras function. Most screens that you will see are action screens.
- To unset a selection, either choose a new default, or Shift-click the same selection. A screen will appear to confirm your action.

Example:

Pressing Shift-click on a radio button at the left in the Libraries & Books screen your selection the 'default' book.

Shift-clicking 'Names' or 'Subjects' or 'Clause-Sets' in the middle box sets the default display in the Clause Selection Screen.



Shift+left mouse click will set any option as the new default.

39.8 Administrative File Locations

Pathagoras works (sometimes simultaneously, usually sequentially) with a wide variety of files. They are stored in locations that after you peruse this article, you should find to be logical. However, without a solid understanding of how the various files work with each other, understanding Pathagoras' file locations strategy will still be a bit daunting to the novice user.

Note: We could 'force' the placement of all files in a particular location. Other programs do that. But one of Pathagoras' great strengths is that it imposes little in the way of where you must (or even 'should') store your files. It works with what you currently have, and let's you change things if you want to modify things later on. We elected not to change that.

The below should provide the user with a good working knowledge of the different kind of files that Pathagoras uses and how their placement on your computer or on your network affects how things are done. Here is a listing of the five 'kinds' of files on which Pathagoras depends.

Administrative Files:

Pathagoras.dot:

This is the actual document assembly 'engine.' The installation routine will determine and then place this file in the startup folder of the version of Word you are using.

The typical path to the Word startup folder is "C:\Program Files\Microsoft Office\OfficeXX\Startup"

'XX' in the above model represents the version number assigned by Microsoft to the copy of Word you have installed. Here are the values: Word2000 = 9, Word2002/XP = 10; Word2003 = 11; Word2007 = 12; Word2010=14).

If you are using a 64 bit computer, the 'Program Files' portion of the address with read "Program Files(x86)".

Other Administrative Files:

The 'administrative' files required by Pathagoras to provide information to the various screens and overlays are located in the folder "C:\Program Files\Pathagoras." (Add "(x86)" to 'Program Files' for 64 bit computers.)

These files are typically simple Word or .txt files that support how the program operates in your specific situation. They include local Help System files, the DemoData (files used to illustrate the program), and certain licensing information. These files are common to all installation.

Also included is the specific information about 'Library and Book' and 'Profile and SmartPath' assignments you have made. These files are the collections of 'pointers' to the various folders which contain your source documents. These pointers are stored as text files.

Instant Database Matter Files

[Other sections](#)¹⁸⁶ of this Manual describe how Pathagoras uses [bracketed variables] as place-holders for personal information. After you have assembled a document and activate the Instant Database system, a screen appears presenting any [bracketed variables] at the left and a space at the right for you to provide 'personal' values to replace those variables.

These variables-to-values pairings can be saved and reused (that is the *Database* part of Instant Database). All such saved pairings appears at the upper right of the Instant Database screen.

But "*Where are these files?*" is the question that will be addressed here.

The initial (default) storage location of Instant Database records is "C:\Program Files\Pathagoras\IDBs". If you navigate to that folder you will see a series of .csv files (csv stands for 'comma separated value') which bear the names that either you or another user assigned, or which reflect demo files and names that shipped with Pathagoras.

If you have but a single license, you probably should just leave the files here.

However, if you have Pathagoras installed on multiple computers, a whole new set of issues arises. If the default location is preserved, each user will be pointing to their 'home' folders. For many customers, that is perfectly okay. Many customers do not want or need to share data with other users. Many customers have office environments such that the client/customer lists in one section are distinct and should be maintained separately from the other sections.

On the other hand, many customers do wish that all users in the organization (or at least all users in a particular section) share a common collection of stored client/customer data. To make that happen, you need only to point each user's IDB records location to a common folder. This is very easily done. [See this page](#)²⁵⁰ for instructions and illustrations.

Instant Database Input Form Masks

An 'Input Form Mask' is Pathagoras version of an Interview Form. A list of such Masks appears at the upper left of the Instant Database screen. These masks are entirely optional. In brief Masks contains the variables used in the particular 'genre' of document you might be composing ('will', 'contract', 'bids and proposals' etc.), but in addition to the variables, the masks have completion tips and examples to guide the users as to how to fill out the Instant Database form. They are discussed in more detail [here](#)²⁶⁰.

Masks are typically stored along with your Instant Database client/customer data records. But they can be stored separately. Some users want to share the company's Input Masks ('don't reinvent the wheel') but don't need, and don't want want, to share the client/

customer data records ('my clients are mine, yours are yours'). That is also easily accomplished. [See this page](#)²⁷¹ for instructions and illustration for sharing Masks independently from Data Records.

*MultiChoice *Lists**

[MultiChoice *Lists*](#)⁴⁵⁸ are powerful tools which allow you to effectively present to a user at document assembly time a long list of options, but you reference those options by a single word. Think "**states**" as representing the 50 United States of America. Click the above link for a full discussion of this feature.

MultiChoice **Lists** can be shared. To share a common list, just make sure that all users point to the same location. Read the text found in the MultiChoice **Lists** section of this manual for more information.

Source Text

Document Assembly Source Files ('Books')

This may be the easiest of all Pathagoras concepts, but sometimes 'easiest' is hardest as new concepts are introduced. You simply need to experience it to appreciate its underlying simplicity. Many other programs cannot operate unless the source text is stored in a specific folder (or sub-folder within that specific folder). Pathagoras has no such limitation. Many Pathagoras users come to us after having used those other programs. We need to rid you of any such preconceived notions. The question "Where are the source clauses stored?" is answered simply "Where you last put them -- and just leave them there."

Pathagoras document assembly modules operate on a *pointer* system. We have elsewhere defined a 'book' in Pathagoras as a collection of files that can be assembled into a complete document. But that definition is not fully accurate. A 'book' is actually just a *pointer* to a folder, not the folder itself. When you call upon Pathagoras to assemble a document, you select a book from the displayed library. When you click the Next button, Pathagoras does not literally 'open' the book and present the clauses within it (as the metaphor might suggest). It merely follows the *pointer* reflected by the book, and, using normal Windows/Word functions, reads the names of the files in the folder. It then presents the document names to you for selection.


It doesn't matter to Word/Windows where you store files. And it doesn't matter to Pathagoras either. Pathagoras can find them regardless of where you store them. So, store your source files on your local system, on your local area network (LAN), on a Virtual Private Network (VPN), on a SharePoint server, etc. It just doesn't matter. Then simply point or re-point the book to that location. It could not be simpler. Read more here on [mapping a book to a folder](#)⁷⁶.

One more important observation. A Pathagoras 'pointer' is just that -- a pointer. *It is not the actual folder or the actual files.* So, if you delete a book, you are merely deleting the letters that make up the pointer. You are not deleting any files to which the pointer points. So you can 'erase' a book without fear that you are somehow erasing or deleting your target files.

Completed Documents

Client/Customer Files

After you have assembled a document, you likely will want to save your work, and you

likely will save it in a location quite 'distant' from the location from which the source file came. As indicated in the paragraph immediately above, source files are accessed by pointers called 'books' which are collected in libraries. Client/Customer files should not be mingled with your source files, and Pathagoras segregates them with an entirely separate document management system called *PathSmart* and *SaveSmart*. Just like books, your client/customer files can be saved anywhere, and the method by which you point to them is a system similar to that of 'books.' That is, you just point a *SmartPath* to the folder which contains the client or customer's files. Click here to [read more about Pathagoras document management system.](#)  534

Networking Features

Common Profiles Path

If you have purchased 3 or more licenses of Pathagoras you are entitled to unlock the networking package that makes sharing of Libraries and Profiles easier.

Let's make one thing clear at this stage. Whether you have one or one-hundred licenses, Pathagoras always operates in any network environment. The networking features discussed here do not enhance a user's ability to recall or save or use documents across any network any easier. Such power is already present in the base program.

That said, picture the following. Let's say that you have created a Real Estate 'Library'. In that Library, you have shelved 10 individual books. It took you all of 3 minutes (it can be done very quickly) but even so, you want to save the other users on your multi-licensed system any time that you can. (Imagine further that you are the administrator of the system. The other users in the office did not participate in the design of the system and they simply may not know where the individual books are located. It's your job to 'set them up'.) That's what the Networking Features are all about. They are administrative in nature, not substantive.

The Networking Features allow you to create a Common Profiles Path and to upload the Real Estate Library to that location. Keep in mind that, as discussed above and in other places in this Manual, a 'Library' is a collection of pointers to the source text, not the actual source text.

The other registered users can then download the Library to their local computers by visiting the Network Files tab in their Utilities/Settings screen. Once you have posted a Library to the Common Profiles Path, you can use the messaging features and send a note to the registered users to download the new/modified Library. (One more reminder: if the *location* of source text is changed (i.e., a new folder), the Library must be changed and users should be notified. If the source text is simply modified or augmented, but the folder location has not changed, the Library -- file pointers -- has not changed. No notice needs to be sent.)

Click there to [read more about creating and using the Common Profiles Path.](#)  557

39.9 System Security

Pathagoras provides tools that will prevent unauthorized users from easily making changes to settings.

There are several levels to this security. Some involve passwording to prevent access to certain features that are displayed to the end-user. Others involve simply hiding certain features

from the end user, suggesting that the feature is not even available.

PASSWORDS:

Two levels of passwords can be set.

- One level excludes all users from making certain changes;
- A second level of security locks out users who do not have 'administrator privileges' on his or her computer.

To enable the password function, call up the Utilities/Settings screen. Scroll down to the very bottom of the Current Settings (front) page of the screen. Click the 'System Password' button.

Note: The type of security that is available will dissuade, but not prevent, operators from going where they should not in Pathagoras. the security provided is not 'fail-safe.' An experienced programmer or user familiar with registry issues could conceivably 'hack' into the file that stores any password you may set. The security is not intended to encrypt or provide 'absolute' security regarding specific documents or folders. (See 'Document Security' for a discussion on that topic.)

END-USER SETTINGS:

Security can be greatly improved against wandering fingers simply by not showing certain program features. Pathagoras provides a feature called 'End-User Settings' which, if enabled, displays only a minimum set of editing choices to the end user. While not primarily designed for 'security' purposes, it certainly can offer this benefit. Click [this link](#) ⁵⁷⁵ to read more about End-User settings (including how to enable the feature.)

39.10 Document Security

While Pathagoras does not provide its own document security features, it lets you easily take advantage of the document security inherent in Word and thereby add a tangible level of document security to your assembled product.

There are many layers to the protection that you can assign to a specific document or to a group of documents within a folder. This is a non-exhaustive list of document security options that are readily available to you:

- Make the document read only.
- 'Password' the document. You can 'protect' (lock) a document against changes by unauthorized users.
- Convert a document to pdf or other formats.
- Prevent unauthorized access to folders and drives on your computer and network.
- Third party providers provide even more enhanced document security (encryption) that will keep eyes off the document or minimally prevent others from changing it.

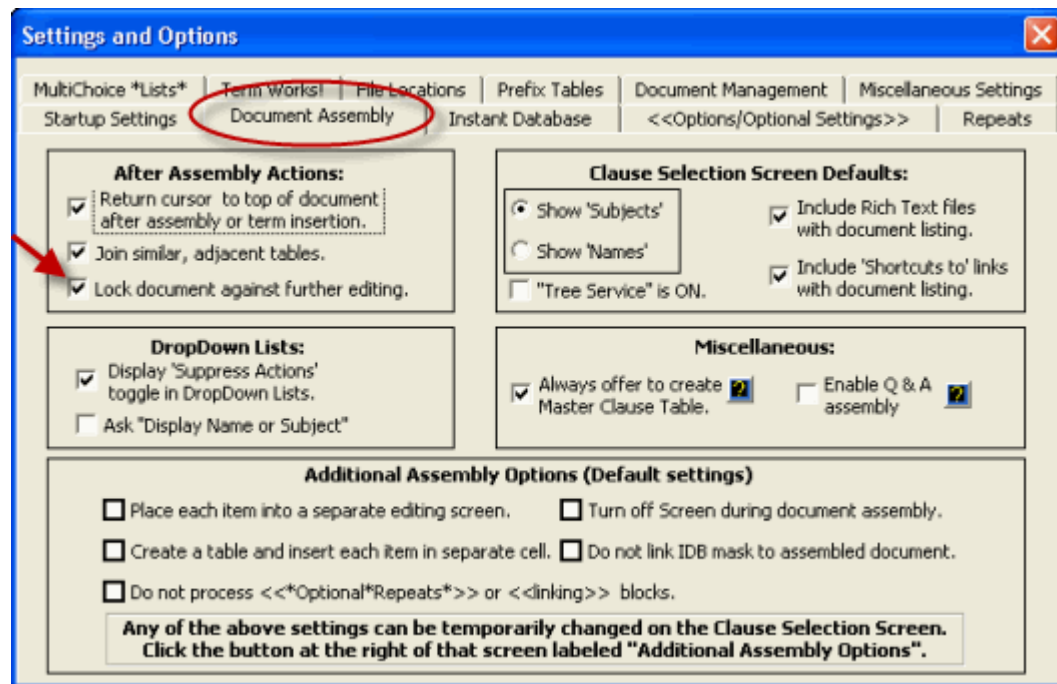
It is beyond the scope of this article to discuss all possible security options. We do, however, invite you to Word's Help system and to the Internet for fuller discussions on the

subject of document security. In those resources you can read about what works, what doesn't, the limitations of each, and techniques by which you can enhance protection for a document, a folder or an entire system against unauthorized access/manipulation.

Enhanced document 'protection' with Pathagoras:

Securing or locking down (Microsoft calls it "protecting") a particular document to prevent unauthorized editing is accomplished via the Tools element of the Word toolbar. To manually protect a document, clicking the "Protect Document" item in the list and follow the prompts.

Pathagoras can enhance this protection by automating it for you. To automatically protect a document after it has been assembled, display the All Settings screen. Click the Document Assembly tab and click the Lock Document box in the upper left.



When you check the Lock Document button, the document will be protected against manual changes. There are 3 exceptions to this:

1. You can insert another clause to the document via the Document Assembly/Clause Selection Screen
2. You can insert another clause to the document via a DropDown List.
3. You can replace [variables] via the Instant Database screen.

When you perform any action described in 1, 2 or 3 above, Pathagoras will temporarily 'unprotect' the document so it can perform the action requested. It then immediately re-protects the document.

Marking specific sections for protection.

Sometimes you may wish to protect only a specified section of your document. Collaborative editing efforts might dictate that easy access be given to certain sections, while other sections need to be 'off-limits'. Pathagoras can help there. Just mark the 'source documents' with "#BEGIN#" and "#END#" (plain text) markers where you want to allow editing. When locking takes place, Pathagoras will scan the document and keep unlocked those sections you have

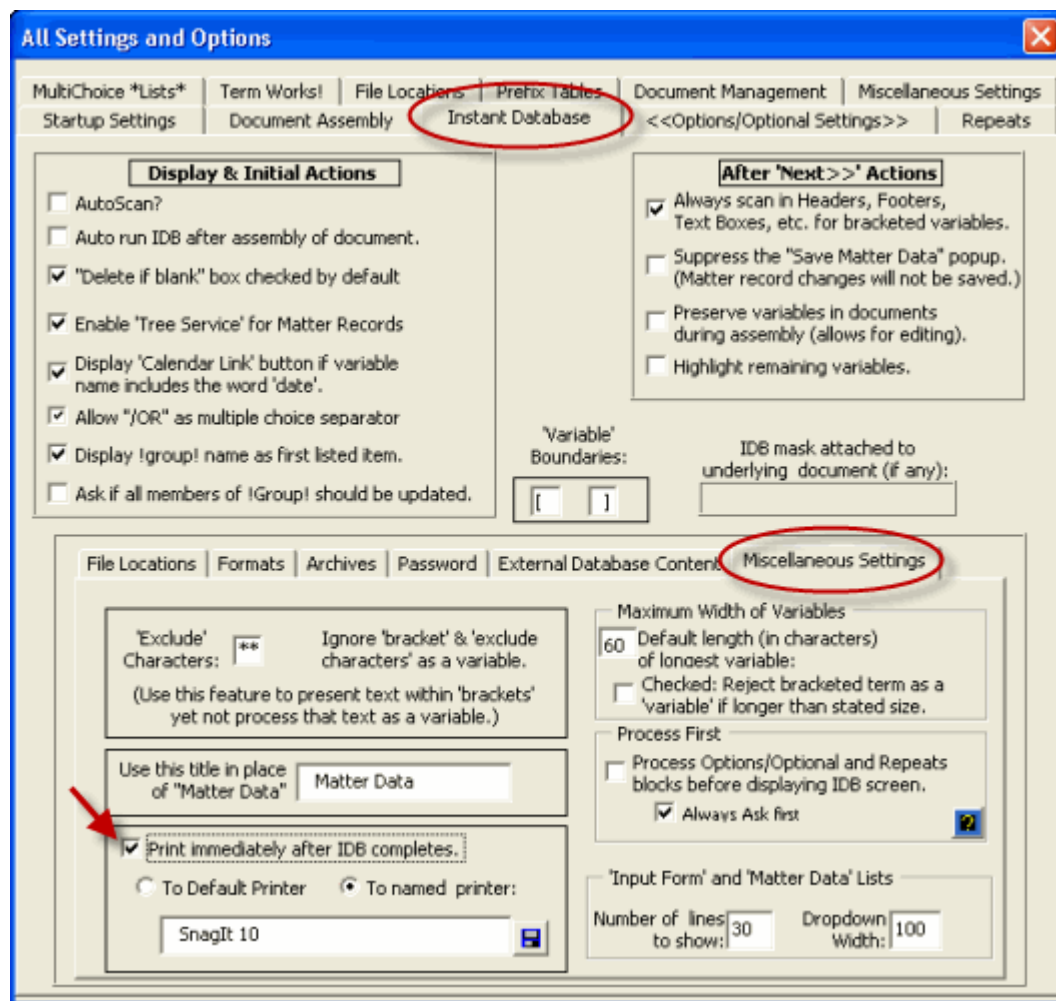
indicated.

CAVEAT: As of this writing, locking is a single level of security which makes it more difficult for a user to edit the document. There is not presently any 'passwording' required to allow editing. Passwording will be added shortly as testing is completed. (So be aware that it would be pretty easy right now for a knowledgeable user to unlock the protected document.) (Even when passwording is added, a password will not be required for the automatic unlocking and relocking described in 1, 2 and 3 above.)

Print after assembly:

This is a quasi security device. Its primary purpose is to automate the printing of your document after you have replaced variables with 'live' values. However, you can designate that the document automatically immediately print to an image (typically a pdf file) after this even, and that way, no further local editing is possible.

To set up Print After Assembly, click Utilities/Settings/All Settings. Select the Instant Database tab. Select the Miscellaneous tab. Select the choices, including designation of the printer to use following an assembly an application of Instant Database data.



=====Extra text. To be removed when editing complete=====

(The document can also be easily unlocked by the user if the user knows the 'protection' password, but that is a standard Word function.)

It will not be possible to insert a clause using Alt-G (and that is only because the document is locked).

As to the automatic print after making the variable-with-value replacements via IDB, I programmed that as well. (The other user was not quite as interested in that.) You can, in the Instant Database settings screen, tell Pathagoras to automatically print the document immediately after replacements are made. If this auto-print is chosen, you are then given the opportunity to select whether the program prints to the Windows 'default' printer or a designated printer (which can be the pdf or any other pre-designated printer).

39.11 Cross-References

If you are writing a long document, you may want to include references from one part to another. Perhaps it is a definition and you want, at the top of the document a line to read "Child, as defined in on page # (or in paragraph #) of this document". Since you don't know precisely where the specific page or paragraph number will be in any particular future document, you will want to insert a cross-reference to be able to return that value after a document is assembled and edited. You can avoid having to manually update pagination, headings or figure numbers by using Word's 'cross-reference' feature. When called, Word will insert a code at the referenced paragraph and a hyperlink at the referencing text.

Pathagoras works well with cross-references. (The ability of Word to track a reference in one paragraph to another paragraph and even though the physical location (and therefore the paragraph number) may vary from document to document, to return the proper reference in the final document.

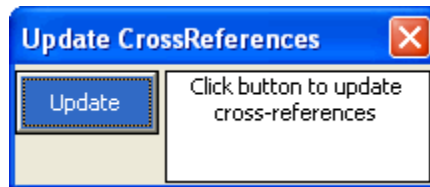
This is true even though the document containing the original cross-references has been disassembled and the presumably re-assembled into a new document. Despite the fact that the documents are no longer stored in the referenced and the referencing documents, when they are put back together, the two can be 'reconnected' easily

Okay, but what about if no cross-reference ever existed, but now I want to create a cross-reference from paragraph A (assembled first in my document selection) to a paragraph that I will assemble later on in the document. Not a problem. However, in order to create the cross-reference, the clauses need to be in the same document at least temporarily. Here is what we suggest.

1. Display the document containing the referencing text. (The one where you want to complete paragraph or page #.) We'll call this document 1.
2. Next, display the document containing the referenced text. We'll call this document 2. Copy and paste that text of document 2 into document 1.
3. Create the reference. If you don't know how to create the cross-reference, do a Google search on "Word Cross-reference" and you will find lots of articles on the topic.

4. A hidden code was added to the referenced text (it resides in the pilcrow character that marks the end of the paragraph). A hyperlink was also added to the referencing text containing a reference. Highlight and cut (to delete it) the referenced text and paste it back into the original document 2. Save the modified document 2. (It is the same text, with the addition of the cross-reference code.
5. Save document 1. It is the same, with the addition of the hyperlink.

Now, when you call in both document 1 and document 2 into an assembled document (regardless of the order or the 'distance' between the two documents, the cross reference pieces are in the new document. After assembly, Pathagoras will detect these pieces and present to you a button to click to update the cross-reference locations.



39.12 Apply 'Wait'

Some computers are faster than others. Some are so fast that an action can actually be called by the program before the previous action has had a chance to finish and register.

To insure that the routines are processed in the proper 'order,' a short (very short) pause needs to be inserted by the program to slow things down just a bit. That pause is invoked with a simple switch. You set it only once, and forget about it after that.

To turn the Apply Wait feature to ON, click the Utilities/Settings entry from the Pathagoras drop down menu. On the Miscellaneous Settings tab, check the 'Allow time to process actions' button.

Don't be too concerned about the 'slow down.' It is less than one-tenth of a second and simply is not a noticeable pause. (Actually, the delay is so insignificant that it doesn't hurt to have this feature set to ON even when you have not experienced processing issues such as those above.)

39.13 Compare to HotDocs® and others

Template development: Pathagoras provides the most straightforward, simplest source document creation process possible.

Most document assembly programs require a tremendous amount of outlining and pre-planning before the first layer of document assembly can be undertaken. While there is nothing essentially wrong with outlining and pre-planning, the assumption is that you already know enough about how the program operates to be able to effectively accomplish the objectives laid out.

Pathagoras likes the concept of pre-planning, but allows the user to experiment with documents under construction without a major investment of time. Fixes are quick and typically obvious and don't involve deconstructing the work done to that point.

The final source document that the document author will confront is laden with fields and codes. Even if the initial builds of your form mimics the 'plain text' approach long ago adopted by Pathagoras, the final, published, documents do not resemble those of the initial builds.

(Hover over a variable in HotDocs® or Exari® and you will see what we mean. Try to edit a variable once it has been converted to a field.)

Sometimes the extension of the resulting 'template' tells you that you are no longer in Word. (Example: XpressDox®, where the template name ends not with .dot but with .xdtpl.) Further indication of the programming nature of the product can be gleaned from the foreign-to-Word ancillary documents that must be created (HotDocs and XpressDox) in order to properly run the program, and the admission that the templates must be 'run' (a sure sign that the template is not capable of 'stand-alone' usage.)

None of these difficult and confusing things exist with Pathagoras.

Here are features you will appreciate:

- You start in Word, end in Word and every intermediate step is in Word.
- Your forms, your variables your optional text blocks are always easy to read. All is in plain Word text. All is 'facial' to the face of the document. (By plain, we don't mean unformatted text. Highly formatted text is very possible. We just mean that Pathagoras uses characters of text that come straight from your keyboard, not fields and codes. By facial, we mean you can see it. Nothing hidden. Nothing in other documents. It's all there and 'in your face'.)
- During every phases of form and template development, your source text is easy to change and easy to work with. That is because they are always Word documents. (I know this sounds redundant, but we have a point to make here.)
- Since the program does not convert your work into coded fields (rather it remains in the original 'plain text' at all times), you can easily change it if you find it doesn't quite work the way you intended.
- Because it is always a 'plain' Word document, you can always take an 'Pathagorized' form home with you to edit it on any computer. You can even edit your work on a computer on which Pathagoras is not even installed. (You won't be able to test it, but you can easily do that later.)
- You never have to 'convert' a form document into a distribution version.
- Pathagoras automates whatever you have at any point in time. It doesn't require that you achieve a state of 'finality' before publishing a template. Publish and use what you have with the full confidence that you can return to the identical form (not a 'published' form that resides in some remote location that you may not recall, and once found, that you have to deconstruct).

So what does this really mean? It means that you can build a form or template as quickly or as slowly and deliberately as you wish. See the section we call [Baby Steps](#)⁶⁷⁸. If all you have time for right now is to convert a few variables in your master document, that's fine. The document is still automated, and you will save at least some time. Later on, add more variables and maybe a few optional text blocks. Keep refining until you achieve the form you want. Pathagoras allows you to grow the form (and to grow your system) as quickly (or slowly) as you wish. At any stage you can use your form in your daily work. It is never an all or nothing proposition.

Interview Wizards: In most document assembly programs, you must construct an 'Interview' form to gather information that will be used to assemble the final document. The process of building the Interview is frequently quite complex. The individual controls on the form must be manually connected to 'results.' 'If / then' statements, branches or loops, must be established so that a particular answer points to a particular piece of text, or if not text then to a logic branch or loop. Once you have programmed the form, you then lock it down and save it out to a special location under a special name.

This is *not* how Pathagoras does it. Pathagoras' approach, to this as well as its other features, is still all 'plain text.' There is no programming required. Pathagoras generates an Interview form from the face of the document being processed. It is all dynamic (based on the content of the current document), not based on a projection of what a document must look like for a particular client.

Data collection: Pathagoras' approach to collecting data is quite different from almost all of

its competitors.

With most document assembly programs, the data entry form must be pre-constructed and 'locked down' before it can be used. So, with other systems:

- The system administrator must know in advance every variable that will be used in the document.
- The system administrator must then manually prepare a form that will display and receive values for each of the variables.
- The system administrator must associate this form with each document for which it might be used.
- If a variable is omitted, or a variable is later added to the document (a frequent and expected occurrence), the sequence must be repeated.
- If a new document has variables not contained within the data entry form, or an existing document is augmented by adding text that contains new variables, it cannot readily be used until the administrator re-writes the form to account for the new variable.
- Adding more variables cannot readily be done by the end-user. It is a system administrator's task. It is not done while the end user is within Word. Rather, the master program (HotDocs®, Ghostfill®, Exari®, etc.) must be called, the source intake sheet must be brought up, edited and re-rendered.
- The sheer number of steps that must be performed (and amount of time that it takes) lead some just to say 'forget it.'
- The preparation of, and the limitations of, the data entry forms required by our competitors are the most frequently cited 'frustrators' regarding the 'other guys.'

Pathagoras has none of the limitations stated above.

- With Pathagoras, no pre-designed data entry form is required.
- Pathagoras can scan any document for bracketed variables and present what it finds onto a completion form. No design work is required by the operator.
- That is not to say that a pre-designed form isn't useful. Pathagoras allows the user to create a form which allows the operator to control the order in which variables are presented for completion, and to provide completion tips and examples. This form is called a 'mask,' and they are discussed below and [at this link](#) ²⁶⁰.
- Creating a Pathagoras mask requires no programming skills.
- Creating a Pathagoras mask is done entirely within Word. It uses the same Instant Database screen that is used in other aspects of the program.
- If a particular document contains a variable that is not in the mask, Pathagoras can readily handle it *ad hoc*. Just press Scan. Pathagoras opens up another line on the Instant Database screen for the variable and provides space for its replacement text, and nobody misses a beat.
- If the user so desires, a variable can be readily added to an existing mask and the revised mask can be instantly saved. No need to open a separate program. It's all at

your fingertips.

Timing of data entry: Most, if not all, other major programs require that you enter a client or customer's information before the document is created. The customer must actually be selected before you are allowed to select the document you want to prepare. This is to allow an unbreakable link between the client and the form, and does have some arguable benefits. Pathagoras finds the 'unbreakable link' concept not compelling, and its consequence too restricting. Pathagoras allows you to build any document at any time, and allows you to input client information at any time.

Specialty vs. General: There are many specialty document assembly programs out there. Estate Planning and Real Estate matters are popular areas. Most of these programs perform on a pre-defined set of clauses (some offer the ability to modify the clauses, some do not) and a data and variable set that is either fixed or offers limited ability to add new variables.

Pathagoras could not be the more opposite. Nothing about Pathagoras limits what you can do, what you can save or what you can assemble. The clauses Pathagoras uses are your clauses. And you can easily modify any of your clause to fit the [bracketed text is an automated variable] style that Pathagoras implements.

Desktop vs. Web based: There is a growing trend toward 'cloud computing' and 'web-based services.' See, for example Dealbuilder and Intellidox. Pathagoras is decidedly desktop based. It can connect to any web server you wish, and all data, clauses, etc. can be drawn from either your local computer, any computer on your network (whether local or wide-area) or 'from the cloud.' (We have lots of customers using DropBox®, SpiderOak® and similar web based storage tools.) But everything is running from your desktop and you do not have to learn new setups to cause the program to function.

From start to finish, the operation looks and feels 'local'. (This 'local' aspect is especially important when it comes to responsiveness. Locally stored documents and locally run programs respond noticeably faster than their web-based counterparts.

Owning vs. Renting: With the advent of web-based services, the trend seems to be toward renting services (even software such as Pathagoras) as opposed to owning a product. With Pathagoras, you own the license (although we have a 'rental' version for those who prefer to go this route). Owning the license is hands-down the less expensive approach, even with potential annual support fees built in. (No software supplier would ever set the retail price of the program at less than about 15 months worth of rentals.) With an 'owned' Pathagoras license, you are done with your purchase. You never have to renew the annual support agreement. It's like Microsoft Word itself. If you purchased, and are still using Word2003, you have yourself quite a bargain in 2011. Same with Pathagoras.

HotDocs® is a registered trademark of the HotDocs Ltd

GhostFill® is a registered trademark of Korbitec Inc.

Exari® is a registered trademark of Exari Group.

Intelledox® is a registered trademark of Intelledox Pty, Ltd.

Dealbuilder® is a registered trademark of Business Integrity.

XpressDox® is a registered trademark of XpressDox (Pty) Ltd

39.14 Compare to Quick Parts

Word's 'Quick Parts' tool is Microsoft's implementation of one of Pathagoras' classic features. Quick Parts was introduced with Office 2007. Pathagoras offered a quick save/quick insert feature since Office '97. We called it 'DropDown Lists' (and still do), but it is essentially identical to what Microsoft is marketing.

The essence of the Quick Picks module (and Pathagoras DropDown Lists) is the user's ability to highlight a desired block of text and add it to the content of a target collection.

Here is where Pathagoras still shines over Microsoft's version:

Sharing: Quick Parts are computer specific. While you can send your Quick Parts to others to load, that's the rub. You can share it with other, but you have to package it up, send it to others via email or other electronic transfer, and the recipient must load it on his or her computer. And if you make changes to your Quick Parts, you have to send the changes to other user in the same fashion. There is none of that with Pathagoras. So long as you and the user have a DropDown List pointed to the folder containing your building blocks, both you and the other user always are working with latest version.

Editing: Editing Quick Parts is a multi-step process. You have to navigate through your Quick Parts dropdown, locate the category in which you saved the item, pick it to copy it to the screen (no direct edit), perform editing, save it back (and as indicated above, if you are sharing with other, send them your entire Quick Parts file.) With Pathagoras, it's simply open the document, edit and save it.

Inserting text: With Pathagoras it is simple two-click process to insert a building block. You also have (unlike Quick Parts) the following options regarding what to do with the building block:

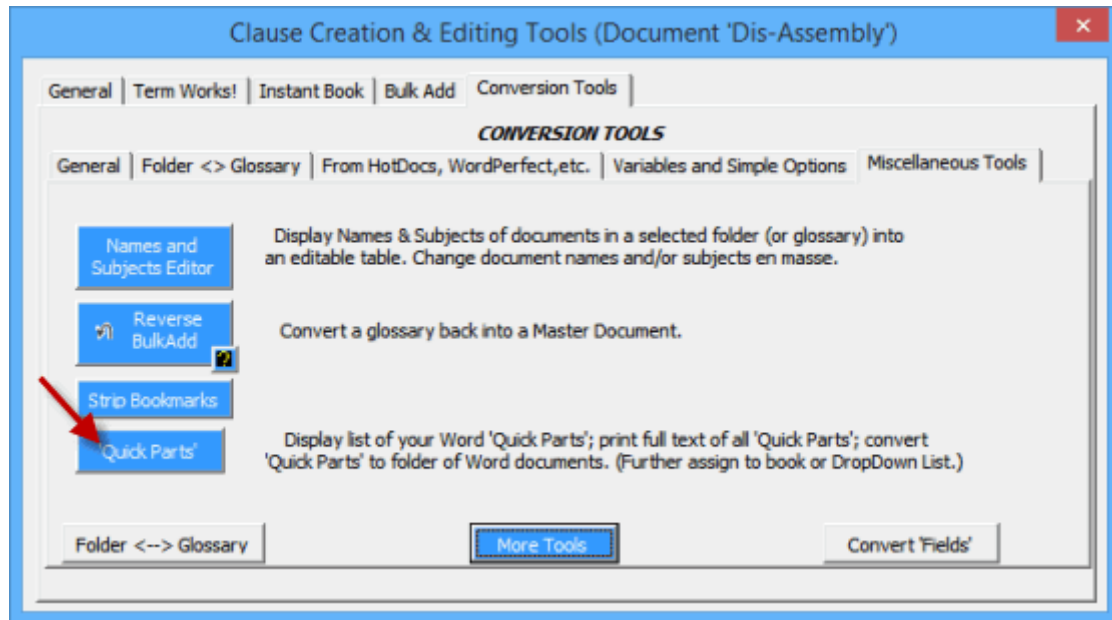
- Insert into current document.
- Insert it as a new document.
- Insert into just your clipboard. (While in the clipboard, the text can be used in multiple locations in the current document, in a separate document, or anywhere else that you can paste text. (Think Outlook, PowerPoint, and anywhere else clipboard text can be inserted.)

Scope: The scope of items to which you can point a DropDown list to far exceeds the scope of items of Quick Picks. DropDown Lists can point not just to Word documents, but to Word Perfect documents, to images, to Excel spreadsheets, PDF files, and more.

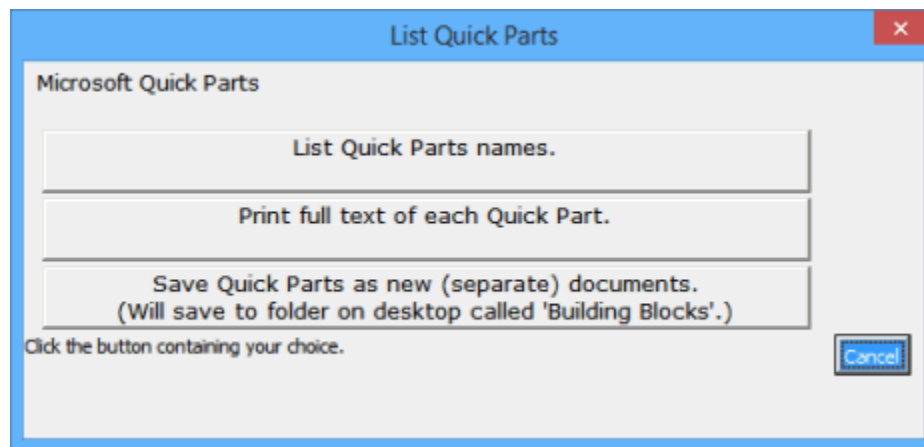
Convert Word 'Quick Parts' to Pathagoras Building Blocks

To convert your Quick Parts to a folder of Word documents ('Building Blocks') you can:

- Do it manually by calling each Quick Pick entry one at a time to your editing screen and saving each item as a document to a folder.(Of course, Pathagoras has this automated. See next bullet.)
- Do it automatically. Select Authoring/Editing tools from the Pathagoras Features dropdown. Select "Conversion Tools | More Tools" and click the Quick Parts button.



A menu giving you choices appears. Consider the first two options for printing lists of your Quick Parts text (so you have a nice printed list if you decide to keep them), but click the 3rd item to perform the conversion.



The Microsoft Quick Parts will be saved individually into a folder on your desktop called "Building Blocks". You can then move the folder to any other location and use it for any purpose. Of course, we at Pathagoras would encourage you to use it as a Document Assembly book or point a DropDown List to it. Your options then become quite limitless.

39.15 Comments

'Comments' are short descriptive passages and explanations regarding any aspect of a document that are stored as one of the document's built-in properties. Comments can be used for any purpose, but most often they are used to explain to others what purpose the document fulfills and the circumstances under which it should be used.

We strongly encourage use of Comments. They can provide great assistance to an end-user who is trying to decide whether and when to use a particular clause. It is one more tool that can make document assembly a more accurate process.

If you 'Preview' a clause (either in the Clause Selection screen or via a DropDown List), Comments (if any) will appear along with the actual text of the selected term.

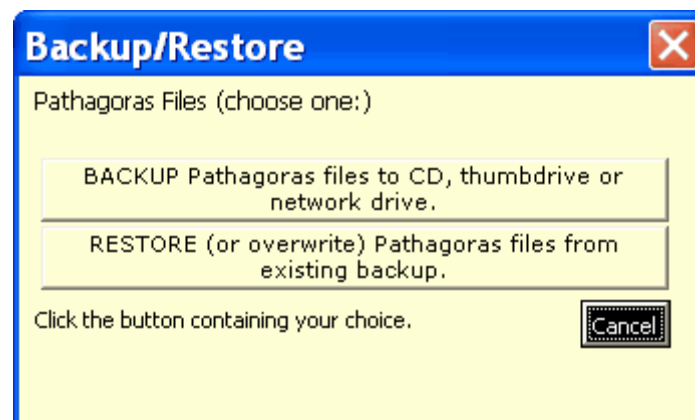
To add a Comment to a document, simply display the document onto your editing screen. Then depending upon your version of Word, follow these steps:

- **Word 2010 or Word 2013:** In the right panel, you should see a list of document properties, one of which will be Comments. Click it. Add/edit the text in the Comments box. (If you don't see "Comments," click the Properties drop-down list (upper-right side of the screen) and choose Show Document Panel. Locate the 'Comments' link. Click it and add/edit text as appropriate.
- **Word 2007:** Click the round Office button. Click 'Prepare' and then click 'Properties'. Locate the Comments box and add/edit the text as appropriate.

Note: The Comments discussed here refer to *Document Level* comments. Word offers text level comments as well. Text level comments (associated with specific portions of the document, including words, sentences and paragraphs) are typically used for collaboration notes, but they can also be used to explain the purpose and usage of a particular section of text. Text level comments will not appear in the Preview screens discussed above.

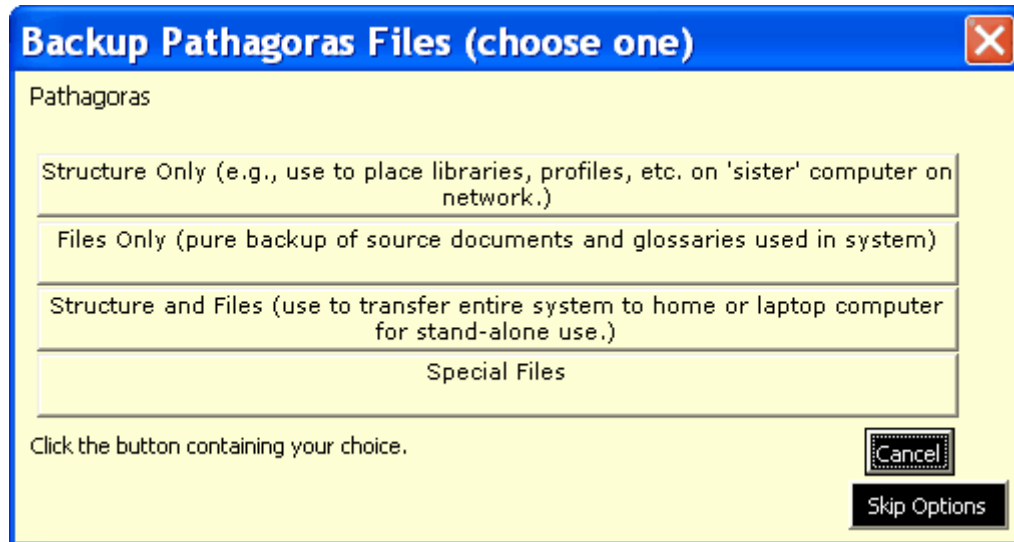
39.16 Backup/Restore

Pathagoras comes with a multi-dimensional backup and restore tool. Access to these tools is via the Miscellaneous tab of the Utilities/Settings screen. You can backup anything from settings (i.e, the library and profile pointers) to every file to which every book in your entire system points.)



39.16.1 Backup

If using a removable backup device, insert the backup media in your CD or thumb drive. Display the Backup Screen from the Miscellaneous tab of the Utilities/Settings screen.



Structure Only: Select this if all you need is to duplicate the Settings (the names of the libraries and books, and the pointers to the various file so that the results can be transferred to a 'sister' computer on the network).

Files Only: Select this if you simply wish a backup of the source forms and other documents that form the backbone of your system. Pathagoras will open every single library and determine the location of the source documents (glossary or folder) and copies them over.)

Structure and Files: Both of the above.

Special Files: Instant Database masks, registry settings, Multichoice *Lists* and DropDown lists.

39.16.2 Restore

Enter topic text here.

39.17 'Plain Vanilla' Paragraph Numbering

'Plain Vanilla' Paragraph Numbering

Paragraph numbering in general was discussed in detail above in the section entitled [Paragraph Numbering](#)³²⁴. The following presents a fully optional 'Plain vanilla' automatic paragraph numbering option.

Sometimes Word's native paragraph numbering routine returns values and formatting that are unpredicted and unpredictable. Trying to correct the result can be frustrating and unsuccessful.

Sometimes Word's numbering routine works well on one computer, but returns different results when you work on the same document on a different computer. This almost always is a styles issue. See Styles.

Pathagoras provides what it calls a 'plain-vanilla' system for numbering paragraphs. The system converts the paragraph to normal text (affirmatively deleting heading and paragraph numbering information inserted by Word in its hidden and untouchable code fashion. Pathagoras then inserts an appropriate, and visible and readily viewable paragraph numbering field as designated from the numbering menu. The parameters from which you can choose include

- (a) numbering styles: 'plain' (7 varieties), outline (2 varieties), or legal
- (b) emphasis: bold, italic
- (c) level: 1 thru 9
- (d) spacing before and after the number.

If you understand and like working with Word's native paragraph numbering system, stick with it. This 'plain vanilla' system is not as powerful as is Word's. But if you like to be able to 'touch' the numbers, and avoid the sometimes unpredictable results from Word's offering, try this.

39.17.1 SavingStyles

You can save up to 5 paragraph numbering styles. To save a style, click the "Save Style" entry under the Emphasis, etc. element of the Numbering toolbar. Pathagoras will ask you to provide a name.

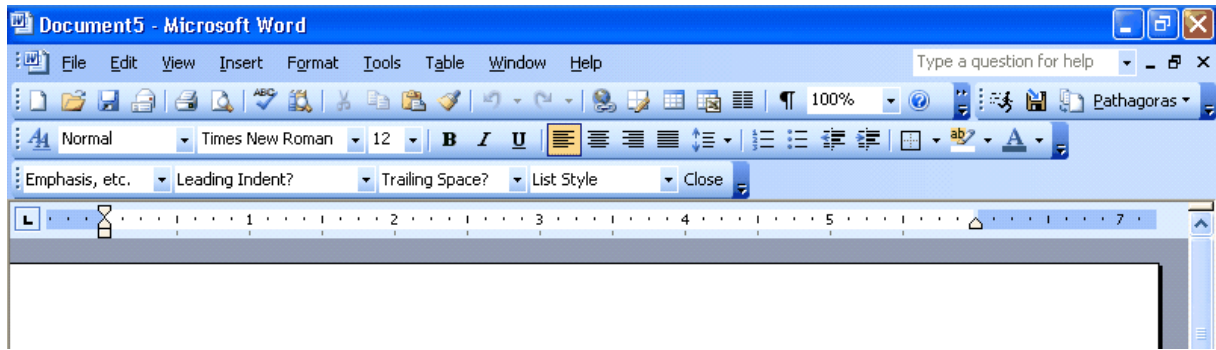
➔ You can also press <Alt-N>, which Pathagoras will recognize as the Save Style command when Numbering is active.



To recall a style, click the "Recall Style" entry.

39.17.2 Using Plain Vanilla

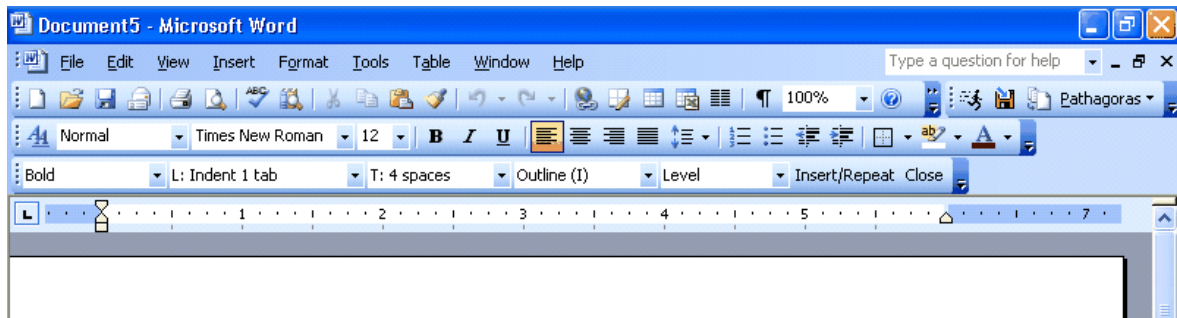
Activate Pathagoras' plain-vanilla numbering system by pressing <Alt-N>. A series of drop-down menus will appear at the top of the screen. You will also be given the option to assign <Alt-1> thru <Alt-9> as trigger keys representing levels in your numbering scheme.



Plain Vanilla Numbering Toolbar, before selections

- (1) Make selections as to emphasis, leading and trailing spaces and the type of numbering ('plain', outline and legal) from the 4 drop-down menus that are displayed.

➔ 'Help' screens are provided at the bottom of each drop down list for assistance.



All selections except outline 'Level' have been made.

- (2) Place the cursor at the beginning of the first line/paragraph to be numbered. Press <Alt-1> (or the Insert/Repeat button) for the first level of numbering.
- (3) Place your cursor at the beginning of the second line/paragraph to be numbered. If this paragraph is a sub-division of the above paragraph, press <Alt-2> (or press Level 2 from the Level menu). If it is the same level as the first paragraph, press <Alt-1> or the Insert/Repeat button. Continue pattern through the whole document/glossary. If you make a level (or other) mistake, just choose the appropriate parameters and repress the trigger key.
- (4) Re-start numbering from "1" (or "I" or "A", as appropriate) by clicking <ReSet to 1> at the bottom of the 'Level' drop down list.
- (5) Close the numbering menu by pressing the <Close> button.

➔ Numbering can stay open constantly. You can save and recall numbering styles. The last style is 'sticky' and will be reapplied when you exit and then return to Word.

➔ As you are numbering paragraphs in a glossary or in documents that are the building blocks for documents-to-be-assembled, the 'automatic' numbers that appear before each paragraph will not necessarily be the numbers that ultimately will appear in the final document.

For example, let's say you have 5 variations of a certain clause, only one of which will

be called into the assembled document. As you number the paragraphs in a glossary, they might show with paragraph numbers 10, 11, 12, 13, 14 and 15, respectively. You know that these numbers will not be appropriate for the final document. Not to worry. When you call in one of those paragraphs during the document assembly process, it will automatically bear the proper number. And previous and subsequent paragraphs will also bear their proper sequential numbers.

The Pathagoras System

Tips and Tricks

Part



XL

40 Tips and Tricks

Enter topic text here.

40.1 Baby Steps

This may be the most important topic of all in this Manual. It deal with the KISS principle, the first three letters of the acronym standing for 'Keep It Simple'.

Pathagoras is designed in such a way that you can keep KISS firmly planted in mind.

When you first begin to implement a project, keep it simple.

- Leave everything in place. You do not have to create a 'Pathagoras folder' to store your documents. (You may if you wish, but that is a step you can take later on.)
- Add only a few variables before you begin testing. Other programs have to convert your documents to their forms before you can test anything, so the tendency is to wait to test until the form is complete. Pathagoras, being all Word, and plain text, you can test immediately with no time wasted on compiling a form.
- You do not have to create new documents just because you have 'Pathagorized' them. Let's say you have a form named "Simple Will." That form has underlines as holding spots for the variables. You manually change the first underline to a bracketed variable '[Testator]'. You then want to save the document. You now have a choice:
 - Save and overwrite the original "Simple Will" document,
 - 'SaveAs' and create a new document.

Our preference is the first choice. When you make a simple change as described above, you have not affected the essential character of the document. It simply has [Testator] instead of _____. It is still your "Simple Will" form and there is no reason to clog your system with yet another Simple Will form.

- Baby steps: In the beginning, don't try to use the high end tools, or the ones you don't yet understand.
 - Use simple square brackets to identify your variables.
 - Use {simple curly braces} to identify optional sections of text (text that you want to keep in some cases and discard in others).

Only after you have assembled a few client documents based on the simpler approaches should you take on the more sophisticated tools. You won't be doubling the work. That's because Pathagoras has a tool that can automatically convert your 'simple' forms to those which will utilize the more advanced and robust tools.

Here is an example:

You have a letter into which you have inserted a series of simple {options} blocks. You assemble the letter 4 or 5 times and realize that several of the options blocks are closely related -- if you choose to keep the first option block, you necessarily want to keep the other closely related ones. So you decide to assign each to the same group by inserting a simple *!groupname!* just to the inside of the option. E

E.g., { !children! I have [number of children] } and later in the document: { !children! I

bequeath my property to my children.}

Still later, as you are assembling more and more documents, you may observe, "it sure would be nice if, instead of answering the variety of questions posed throughout the document as individual questions, I could answer them all at the beginning of the assembly process."

And Pathagoras can do that as well for you. From something as easy to create as the {simple options} discussed above, Pathagoras can automatically create for you robust <<*Options/Optional*>> text blocks. These blocks will allow you to pose more meaningful questions and prompts to the end users, all with plain text setups, of course. From there, Pathagoras can automatically create a 'question menu' at the top of the document. Pathagoras does so by scanning your document for <<*Options/Optional*>> blocks, by 'dissecting' out its component parts, and creating what we call an Ask Table. And, true to Pathagoras underpinnings, the Ask table is plain text and facial, and very easy to edit further (refine the text of the questions, change the order of appearance of the questions, etc.)

All of the above is done within the existing Word document, and none involves any real duplication of effort. Because you took 'baby steps' to accomplish what is quite a sophisticated end, you hopefully understood what what happening at each step along the way. So if you need to augment the now 'powerful' document, you are not overwhelmed by the now robustly Pathagorized document.

40.2 Captions and Signatures

The caption of a legal pleading (frequently called 'style of the case') and signature blocks present challenges for document authors. There can be the 'easy' single party pleadings (one plaintiff, one defendant) or there can be any number of combinations of multiple parties (one plaintiff, two defendants; two plaintiffs, one defendant; two or more of each, etc.) You would like Pathagoras to be able to 'design' the proper pleading based on your feeding it the proper number of parties to the litigation. (Note: this 'problem' is not limited to parties to a lawsuit; it occurs in contracts involving one or more Buyers and Sellers; Shareholder agreements with multiple combinations of Shareholders on each side; etc.)

There is no universal solution, but we offer a few tips

If you have a finite number of possibilities (i.e., never more than two of each), you can create each variation within the body of the document itself and create an options block, separation each possible caption from the next with a "/".

<<*Options*!Parties!One Plaintiff, One Defendant/Two Plaintiffs, One Defendant/One Plaintiff, Two Defendants/Two Plaintiffs, Two Defendants*

(caption1 text here)/

(caption2 text here)/

(caption3 text here)/

(caption4 text here)>>

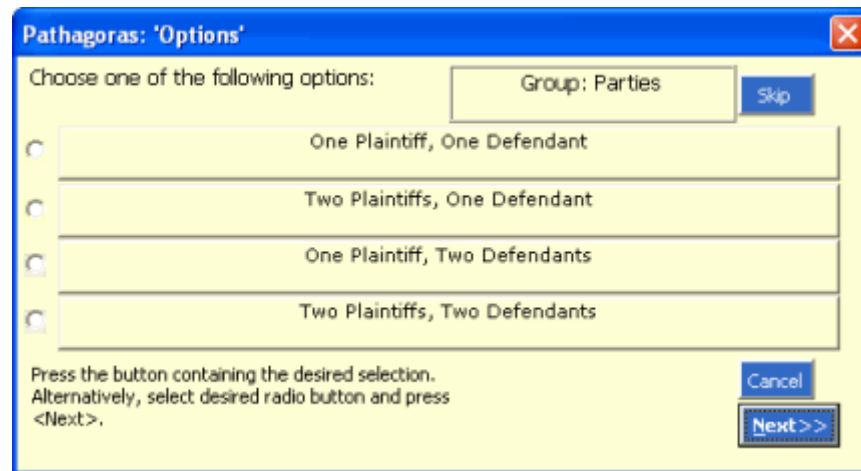
(Remember from previous lessons regarding Options text, the coloring in examples such as the above and below is provided for emphasis only. It is not required for any functional purpose.)

Alternatively, and we like this better, you can create each variation as individual documents and store them under appropriate names in the SuperFolder along with other 'universal' documents that can (and should) be stored there. Maybe the names of the various captions will be P1D1.doc; P2D1.doc, P2D2.doc, etc.

In the base document itself, create an options block that reads:

```
<<*Options*!Parties!One Plaintiff, One Defendant/Two Plaintiffs, One Defendant/One
Plaintiff, Two Defendants/Two Plaintiffs, Two Defendants*<<P1D1>>/<<P2D1>>/
<<P1D2>>/<<P2D2>>>>
```

When a document that contains either of the above <<*Options*>> blocks is processed, it will present the options in the below fashion:



When you make a selection, Pathagoras will delete the unwanted captions, and leave the desired one. (In the first example, the text itself is what remains. In the second example, after the initial processing of the <<*Options*>> block is complete (and let's assume that the second choice was selected, the term <<P2D1>> is what remains. It being a double angle bracketed term, Pathagoras reprocesses it and determines that a document is being sought. Following the Search Order Rules, it finds the document p2d1.doc in the Super Folder and inserts its contents at the proper location.

When the number of choices exceeds just 'a few,' consider using the [<<*Repeat*>> function](#)³⁸². Be sure to read the sections regarding [Repeats with !Groups! and incrementing values](#)³⁸⁴

Signatures: All concepts and above examples apply with equal weight to constructing signature blocks. Only the location in the document is different. Don't forget to use [!Groups!](#)¹⁵¹ to tie the response made at the beginning of the document processing to the signature blocks.


See more legal caption/style examples using the Repeat function [at this link](#)³⁹⁹.

40.3 Capturing Data Independently of Document

You likely have observed that the typical order of a document assembly session in Pathagoras is to create the document and then capture the personal data via the Instant Database. If the personal data exists from an earlier document assembly session, then that data can be reused.

But what if you want to capture the data first. In other words, as soon as the client/customer

or patient registers, a non-author would input the data into the Instant Database. That is entirely possible and we offer 3 scenarios.

Create a Mask. This is fully discussed starting in [another section of this Manual](#)  ²⁶⁰, and you are invited there. If you assign that Mask to one of your books, the person assigned to inputting data can activate the mask from the Libraries and Books screen that appears when the Document Assembly icon is clicked and the book selected.

Create an Intake Sheet: Another alternative is to create a document designed especially for capturing data, and nothing else. This might be called a "Case Information Sheet". It contains all of the variables you are likely to use with documents subsequently created. This type of document serves a dual purpose. You (1) capture the data at the outset and (2) create a printout of that data for reference purposes in one fell swoop.

(Be mindful, even with the Case Information Sheet approach, you can either run a scan on the intake sheet or call up a mask from the Instant Database as the guide for completing the sheet.)

Case Information Sheet

Client's Full name: [Client Full Name]
Client's Last name: [Client Last Name]
Client's Salutation (for correspondence): [Client Salutation]
Client's Street address: [Client Address 1]
Client's Street address2: [Client Address 2]
Client's City, ST ZIP: [Client City, ST ZIP]
Short Matter Name: [Matter Name]
Matter Number: [Matter Number]
Court Case Number: [Court Case Number]

Pathagoras for Excel:

Pathagoras' separately distributed Excel module can be used to create a data collection screen. This is fully discussed in a [separate manual](#), but in a nutshell:

- With Pathagoras for Excel, you can create an intake sheet using the powerful design features of Microsoft® Excel. You can post the sheet on the web or send it to the client/customer/patient via email for completion and return. (You can also post this on a computer in your office, put your client in front of the screen and have the client complete the form electronically.)
- Upon its return, Pathagoras/Excel can create an Instant Database record from the Excel spreadsheet as if you had completed an Instant Database form in the 'standard' fashion.

The Pathagoras System

Upgrading Issues

Part




XLI

41 Upgrading Issues


This section discusses 4 main topics:


- Upgrading Pathagoras when a new version is released, including how to check to see if you are using the latest version.
- Upgrading to a new version of Word, and the steps you must take to insure continuity.
- Upgrading your operating system, and the steps you must take to insure continuity.

 Pathagoras does not automatically upgrade to the latest version of the program (many don't want the latest version), or notify you of the updates every time you start up Pathagoras. That would drive you crazy. But we do publish periodic newsletters announcing major and minor updates, and we make it easy for you to determine your upgrade status. If you decide you want the update, we also make it easy for you to download and install the latest version of the program.

41.1 Upgrading Pathagoras

Pathagoras is always changing. It is upgraded on a fairly regular basis as users offer suggestions to improve current features or to add new ones. It is this reason why Pathagoras is distributed only over the Internet. (No CD could ever be 'current.' As soon as one is mailed, we might have implemented a change.)

To upgrade Pathagoras, follow the steps outlined [at this link](#)  ²⁶.

 Upgrading Pathagoras is a simple, painless process:

- Pathagoras preserves all libraries, books, settings, Instant Database records, etc. You will lose nothing when you upgrade.
- You do not have to uninstall a previous version of Pathagoras to upgrade.
- The transition from a demo version to a licensed version is seamless..

 **Network users:** follow the special [upgrade instructions](#)  ⁵⁶⁴ to most efficiently upgrade Pathagoras in a network setting.

41.2 Upgrading my system or adding new users.

I have a new computer. I have reinstalled Pathagoras. What else do I need to do to reinstate Pathagoras to the same setup that I had on my 'old' machine.

or

I am adding new users to my Pathagoras network. I want to point the users to the same Instant Database records, MultiChoice *Lists* etc. as I have on the older installations.

Many users become anxious when they have to add a new computer to the network or switch computers to a fancier model. The situation is made worse when the change is involuntary (such as a crash and complete loss of data).

This outline is intended to get the Pathagoras side of your operations up and running as soon as possible.

Whether an upgrade, an expansion of the network, or the recover from a crash, the approach to installing/reinstalling Pathagoras, and restoring your files, will be similar. Keep in mind the following general concepts:

- Your source files are, and always were, standard Word documents. They are, and always were, stored in standard Word/Windows folders.
- Pathagoras does not 'own' any of your source files. It simply points to them. As a document assembly engine, it provides 'administrative support' in order for you to quickly locate and assemble documents, but its processing is strictly 'back end'.
- So with the above two concepts in mind, you need not have Pathagoras foremost in your thought process as you make decisions regarding adding a new, or restoring a 'crashed', computer. It requires no predetermined setup. Pathagoras works from files wherever you decide to put them.

So, when you add a new computer to your network, or restore a system after a crash, you need only restore your substantive clauses and documents to a location where you know how to find them. (Of course, it likely will be easier to maintain for the new or restored computer the same configurations you had on the old computer, but that is true irrespective of Pathagoras.)

Once you have put the new computer on line, or restored the old computer from a crash, (re-)install Pathagoras. Then point the various Pathagoras element (books, Instant Database records, MultiChoice *Lists*, etc. to the location where the files actually reside. The checklist on the following page of this Manual may be of help.

If you adding a new computer to your network and want to point to the same locations where other existing computers now point, just open up the setting screens of the existing computers and see where they now point. Then just mimic those settings. (Of course, if the files are not in the same relative location, then you will have to repoint your books to the new location.)

If you own a single license, and are transferring your license from one computer to another, you can use the Backup features of Pathagoras to copy your setup files to (let's say) a thumbdrive, and use the Restore features to move them onto your new computer. The assumption here is that the 'substantive files' for you document assembly system are in the same relative locations on your new computer as they were on your old computer.

Definitions:

Substantive files: Here, we mean your source documents, the letters, clauses, forms and other text that you assemble when you invoke Pathagoras' document assembly features.

Administrative files: These are the help files, dictionary files and other files stored in the "C:\Program Files\Pathagoras" folder, including the 'pointers' for your collections of Libraries and Books, SmartPaths and DropDown Lists. Other files can be stored here as well.

Instant Database files: These are by default stored in "c:\program files\Pathagoras\IDBS". However, many networked users choose to store these files on the server or other common location. Be sure to repoint them to the server if such is your case.

Registry settings: Just like with other programs, Pathagoras stores a myriad of 'default' and 'user-personal' settings in the registry. You normally do not have to worry about what Pathagoras stores there, but just know that when you check a box or radio button to tell Pathagoras what to do on a long-term basis (a 'default'), your choice is recorded in the Registry.

Repointing:

Check out [Pointing and Repointing](#)²¹.

If you have the Networked version of Pathagoras, the process of adding and changing computers is made much easier, but does require the use of (and creation of, if it doesn't now exist) the Common Profiles Path ('CPP').

The process is easy:

- (1) install the program on the new machine from the original source.
- (2) connect to, and register with, the Common Profiles Path ('CPP'). Then, download the profiles and libraries that hopefully have been previously saved there.

Note: If a CPP has not yet been set up, you should so now. That way information can be uploaded from your old machine to the CPP before the switchover is made.

- Using the CPP, the data from your old computer can be preserved by uploading it to the CPP.
- Once the new machine is put on line, download the settings *from* the CPP and be up and running.

Setting up the CPP is done following the steps on this page of the Manual:

http://www.pathagoras.com/help/index.html?enabling_the_network.htm

If the CPP has not previously been created, you will need a networking unlock code. Call or write us for that information if you do not still have it. (It was provided in the email that accompanied your initial purchase.)

If the CPP has been created, but the latest profiles and libraries not stored there, just go to the Utilities/ Settings screen, click Network Files and then upload the Local Profiles and Libraries to the CPP.) Then, after the new computer is 'up and running,' perform steps (1) and (2) above.

41.2.1 Upgrading Checklist

99.9% of all issues that arise as a result of upgrading can be attributed to a simple failure to adjust a file or folder pointer. The main thing you want to keep in mind when upgrading is that Pathagoras 'owns' none of your files. It merely points to them.

Unless otherwise noted, the steps below should be performed regardless of whether you are upgrading or adding a new user. (It is assumed that the computer that is to

receive Pathagoras already has a valid copy of Microsoft Word installed.):

- ☐ If upgrading to a new computer, copy from the 'old computer' to the 'new' all folders which contain documents you regularly use with Pathagoras. We recommend that you keep them in the same relative location. This makes the transition easier. This is not, however, a 'requirement' and you can place the documents in any folder desired.
- ☐ If upgrading to a new computer, copy from the 'old computer' to the 'new' the administrative folders that are unique to Pathagoras. These are typically located in the folder called: "C:\program files\Pathagoras" (If a 64 bit computer, the files will be located in the "C:\program files (x86)\Pathagoras" folder.) We recommend that you copy the entire folder 'intact' and place in into the same relative location on the new computer.
- ☐ Install a 'full license' version of Pathagoras to the new computer. (An upgrade or a beta version that you may have earlier been provided will not work. You will receive error messages. Contact Pathagoras if you need to obtain a 'full license' version.)
- ☐ Check the pointers to your various books in your Libraries. Adjust if needed.
- ☐ Check the pointers to your Instant Database records. Adjust if needed.
- ☐ Check the pointers to your MultiChoice *Lists* files. Adjust if needed.
- ☐ Recreate your DropDown Lists.

With all settings checked and confirmed, you should be ready to roll.

41.3 Annual Support Agreement


In addition to the basic program, your purchase of a Pathagoras license entitles you to the following at no additional cost:

1. all upgrades to Pathagoras issued during the 365 days following your purchase (or conversion of your subscription to a full license)¹.
2. technical support for issues related to the program. You will find our definition of 'technical support' to be quite broad and liberal. If you have a question about the operation of a feature, or if the program (or a document you have 'Pathagorized') doesn't work the way you think it should, give us a call, or shoot over the document you are working on. We'll debug the problem and offer some helpful tips along the way.

If the nature of your question or issue is beyond the scope of Pathagoras, we'll let you know, but we don't mind being called.

To check to see if your Support Agreement is current, click the Pathagoras features button. Click Main Menu button and then the 'About' tab. Click 'Your Upgrade Status'. If your Annual Support Agreement has expired, you will be given the option to renew it. Renewals run from the date of your renewal (and not 'back dated'). So you get a full year of support from the date of the renewal.

Beginning one month prior to your expiration date, we will send a reminder that your ASA is about to expire. You can renew at that time as well. (In this case, the renewal will run from the anniversary date of your original purchase or last renewal of your ASA.)

 Pathagoras does not force you to renew, and your currently installed program will continue to run even if you do not renew. You must affirmatively initiate the renewal process. We do not have any automatic renewals.

¹ If you are a 'monthly subscriber' to Pathagoras, you are always entitled to support and to the latest version of the program. There is no additional fee for either. Just keep in mind that upgrades are not automatic. You must affirmatively initiate the renewal process.

The Pathagoras System

Support

Part

XLII

42 Support

42.1 Customer Service

Pathagoras prides itself on providing prompt, useful and personal customer service. While we hope that this Manual and the other instructional materials are helpful, you can still count on the 'personal touch' of Pathagoras' customer service as a 'first line' of help as well.

We truly enjoy hearing from our customers and potential customers. While we cannot promise that you will never receive a voice-messaging service if you call us, more likely you will receive a live person at the other end.

But if you do get our voice mail, just leave a message. We will call you back promptly.

Contact information is spread across as many places as we could find so that you do not have to hunt for an email address or telephone number, and we repeat it here. Let us know if we can ever be of service.



Pathagoras
Innovative Software Products of VA, LLC
Roy Lasris, President

E-Mail: info@pathagoras.com *(I personally read all e-mails sent to this address!)*

Website: www.pathagoras.com

telephone #s: +1 866-PATHAGOras (1-866-728-4246) (tollfree)
 +1 (757) 877-2244 (USA) (direct line, not tollfree)
 +1 (757) 898-7374 (evenings, weekends)

Address:
117 Chisman Landing
Seaford, VA 23696 USA (I read all mail)

Reporting a 'bug':

We ask you to report any and all program 'bugs' you encounter. This includes 'suspected bugs.' If you have had a problem, do not hesitate to tell us about it.

- Most error messages that Pathagoras and Word generate are rather generic in nature, and most of the time not helpful in pinpointing the precise source of the problem. If you receive an error message while Pathagoras is in operation, and can duplicate it at will, please send us a list of the steps that will generate the error. If we can duplicate it on our end, we can come up with a much quicker fix.
- We would also greatly appreciate your sending any screen shots that may help us to better identify the problem.
- Sending us actual text of documents that didn't quite work would be good too. (We will not share your documents with anyone. We understand the proprietary nature of your personal or business work.)

Remote Assistance:

If you call us for guidance, or to report a bug that can be best explained visually, we likely can most effectively address the situation via 'virtual presence.' In other words, we can (and would like to) virtually sit at your machine. That way we can see the same thing that you are seeing at the same time that you are seeing it.

- This can be readily accomplished via a wide variety of remote assistance tools.
- Our preferred meeting tool is GoToMeeting®, a product of the Citrix Corporation. It is easy to use, offers quick response time and allows us to sit 'together' while we share information on your computer.
- If you feel a virtual meeting would be helpful, don't hesitate to say "Can we do a GoToMeeting?" (If you are otherwise covered under the Annual Support Agreement, there is no charge for the meeting. And don't worry about costs on our end. We have none beyond the annual subscription that we pay for GoToMeeting.)
- To activate remote assistance after we have agreed to a session, display the Utilities/ Settings screen and click the Miscellaneous tab. Click the red "Remote Assistance" button. Enter the Meeting ID number that we will provide in the text box and click the Remote Assistance button again to make the connection.
- If you have a Remote Assistance program that you would prefer that we use, just let us know.

GoToMeeting® is a registered trademark of the Citrix Corporation
--

42.2 Non-technical Support

Pathagoras Author & Staff For Sale!

I am for sale! Yes, me personally, and my staff, too.

We are available to help you set up Pathagoras, establish your Pathagoras network if you have multiple licenses, create or refine your books and libraries, etc.

- **Let us Pathagorize You:** If you simply do not have the time or the energy to Pathagorize your forms and create a system, we can handle it all for you. We have skilled and talented 'Pathagorizers' on staff who can quickly respond to your every request.
- Send one or more of your current systems. We will set up a Pathagorized model system in return. What you will get back is a complete, immediately usable, book. We will also send you instructions on how to place it onto a new or existing library shelf. Since the returned material will be a collection of standard Word documents, you will be able to augment or freshen the text on your own as needed.

Our rates are posted on the website, or call us for a quote. Your savings in future document assembly time will more than recapture the investment.

(If you want to take us up on this offer, we suggest that you send us just one of your systems, not all of them. When you get it back, study what we did and how we did it. Hopefully, then, you will feel so confident in how simple and easy it really is that you will attempt to 'Pathagorize' the next system yourself.)

- **Private Lessons:** If you choose not to read the manuals (I hate reading manuals too), you can simply 'buy' me to provide extended lessons and some 'on-site' (via GoToMeeting) guidance on how to get set up and fully operational with Pathagoras. Put your whole office staff in front of a computer and a speaker phone and we will be all set. And when you consider how far I can get in that time, that can be quite a deal for you.
- **"Will you travel?"** Heck yes! I would love to! (Get me out of my law office, please!) Of course, the airplane ticket and accommodations would be on you. My on-site charges are a bit higher, but the work that I could accomplish in that day or two (setup and training) should make the investment very worthwhile.

43 Help System Improvement Project

Please help us to make this Help System the best it can be.

- If you find an error, please copy a snippet of the text containing the error and shoot it over to us in an [email](#).
- If a topic that you would like to see is missing or misplaced or not indexed appropriately, please [let us know](#).
- If you were looking for a piece of information under a name or topic where you thought it might be and did not find it, [please write](#) so that we can add that link.
- If you have any suggestions whatsoever, [please write](#).

Thanks.

44 Uninstalling Pathagoras

If you need to uninstall Pathagoras, do so from the Control Panel. Select the Add/Remove programs item. Locate and select the Pathagoras entry and click "Remove." **NOTE:** Be sure to exit Word before trying to uninstall Pathagoras. (This is an important first step. If Word is active, so is Pathagoras, and it will not uninstall while active.)

There is also an Uninstall button under the Start|Programs|Pathagoras entry.

If Pathagoras does not uninstall via the standard method, you can manually uninstall Pathagoras by deleting the "pathagoras.dot" from the StartUp folder. Here are the steps:

1. **Exit Word.** (This is an important first step. If Word is running, so is Pathagoras, and it cannot be deleted while it is running.)
2. Navigate to your Word Startup folder.
 - If Pathagoras was initially installed after June, 2013, your Startup folder most likely is:
C:\Program Files\Microsoft OfficeXX\Startup (Office 2007 and 2010) or
C:\program files\microsoft office 15\root\office15\startup (Office 2013 and 365)

- On installations before June 2011, the Startup folder is either:
Word 2003:
C:\Documents and Settings\your name\Application Data\Microsoft\Word\Startup
or
C:\User\your name\AppData\Roaming\Microsoft\Word\Startup

3. Delete any 'pathagoras.dot' and 'pathagoras2007.dotm' files present.

If you cannot find 'pathagoras.dot' in the manner described above, use Windows 'find' or 'search' tools to locate the Pathagoras files. Follow these steps:

1. **Exit Word.**
2. From your main Windows screen, press the 'Start' button.
3. Press 'Find.'
4. Press 'Files or Folders.'
5. Type 'pathagoras.dot' in the 'Named' field.
6. Make sure that the 'Look in' field is set to your C: drive, and 'Include Subfolders' and 'Include System folders' are checked.
7. Delete any 'pathagoras.dot' and 'pathagoras2007.dotm' files that are displayed.

After you have deleted 'pathagoras.dot' (and 'pathagoras2007.dotm' if running Word2007/10) from your computer, finish the manual uninstall process by locating and deleting the directory called "C:\Program Files\Pathagoras". This is the folder that the installation

routine would have created 'by default.' If you did not accept the default locations, you must navigate to the folder into which you saved the Pathagoras source files and delete that folder.

45 One Page Quick Start Guide

Insert variables into document for automated replacement.

Open the document that you wish to 'Pathagorize.' We will call this document the 'source' (original) document. Put [square brackets] around each word or short phrase that you want to serve as a 'variable.' (A variable is a 'holding place' for what later will be personal text. E.g.: [Client Name].)

Create multiple-choice variables in a similar fashion. Listing all choices within the square brackets (as above), but separate each choice with a forward (/) slash. (See example below.)

Enclose text that you want to be "optional" within {curly braces}. (See example below.)

Close and save the document into an appropriate forms folder.

Hints & Tips:

The variable should have a 'variable sounding,' reasonably descriptive name. "[Date of Letter]" is more meaningful than just "[date]". Use "[Name of Customer]" instead of plain "[Name]".

Don't save source document with .dot (template) extensions. Use standard .doc extensions. 'Plain' documents are much easier to work with.

Example:

This is an example of 'Pathagorized' text:

Dear [Customer Name],

Thank you for your order placed on [Date of Order]
for [quantity] [red/blue/assorted] [Product Name].

We will ship your order within 5 business days.

{Please note. Due to the special pricing of these
items, all sales are final.}

[Sales Representative Name]

Assign data to variables

Open or create the document that contains variables. Press Alt-D (for 'Database') to display the Instant Database screen. (It is called Instant Database because you will be able to instantly save the variable to data pairings to a database for reuse with other documents.)

Type the appropriate value into the textboxes for each variable. If you have created multiple-choice variables, select from the drop-down lists. Press the Tab key to move to the next variable.

You may have several pages worth of variables. Press the numbered buttons at the top of the screen to move from pages to page. Press the '>' button to display addition pages beyond 4.

Press the 'Next' button when finished. You need not complete all variables. You can run Instant Database at a later time to fill in the remaining variables.

Libraries & Books

You should never use the original source text for creating a personal document. You should only use a copy of the source text.

Pathagoras makes it easy to implement this rule and always call up a copy of the source text. Pathagoras' 'Libraries & Books' and 'DropDown List' modules allow you this protection in a simple, but very elegant style. Read more about these tools in the [Document Assembly](#) ⁴² pages of this Guide.

Creating a Document

To create a new document:

1. Click the Document Assembly button.
2. Click the radio button on the line of the book from which clauses are to be drawn.
3. Click 'Next'.
4. Choose the clauses/documents you wish to assemble from the left column and move them to the right.
5. Click 'Next'.
6. Press Alt-D to bring up the Instant Database screen.
7. Press the Scan button to read the variables.
8. Type a value to replace the variable.
9. Click 'Next'.
10. You are done!

Index

- " -

"/" vs. "/OR" 146
 "?" 26
 "+" sanitizing 126

- . -

.doc vs. .dot 308

- [-

[**a/an] 127

- { -

{Optional}
 Compare to <<Optional>> 167
 Convert to <<*Optional*>> 168

- < -

<\$>Currency Function (IDB) 239
 <%> Percent Function (IDB) 241
 <<*Ask*. . . prompts
 Elements 501
 <<*If*. . . prompt 506
 <<*Set*. . . prompt 508
 <<document name>> call 104
 <<Double Angle Bracket Functions>> 104
 <<Filename>> Function 244
 <<Folder>> function 228
 <<Optional>>
 Compare to {Optional} 167
 <Alt-D> 186, 590
 <Alt-G> 94, 550, 590
 As a typing expander 592
 <Alt-N> 590
 <Alt-Q> 180, 182, 590
 <Alt-S> 348
 <Alt-V> 122
 <F>ormat Function (IDB) 242

<Fr>actions function (IDB) 242
 <MailTo:....> function 246
 <P>aragraph Function (IDB) 245
 <S>pell-out Function (IDB) 238
 <URL:....> function 247

- 7 -

7-day Plan 26

- A -

A/An
 The Intelligent A 127
 A/K/A List 127
 Acrobat
 As intake form 644
 Completing Acrobat fields 644
 Actors and their Roles 231
 Adding Computer 684
 Adding Records to IDB from External DB 297
 Adding source clauses 306
 Adding terms
 DropDown Lists 359
 Manually (folder of terms) 311
 Manually (glossary) 313
 Term Works! 311
 via Libraries & Books screen 308
 Administrative Files 657
 Administrative Text 144
 Options/Optional 162
 Syntax 162
 Adobe Acrobat
 As intake form 644
 Completing Acrobat fields 644
 Data scraping 648
 IDB variables 645
 PDF Assembly 651
 Scanning for variables 647
 Advanced Array
 Instant Database 218
 Age Calculations 199
 Age math 227
 Alias *Lists*
 and !Groups! 476
 Cascading 468
 Creating 460

Alias *Lists*

Definition 458
 Embedding 488
 Excel 466
 Location 486
 Moving 486
 Re-pointing 486
 Sharing 489

Alias Table

(Embedded) 488

All or Nothing -- not 20

All Settings Screen 576

Alphabetize

Instant Database 210
 Instant Database Variables 268
 Variables 210

Alternative Approaches 622

Alt-Q 322

Alt-Space 322

Analyze ranges of values 503

Anatomy

Optional Text Blocks 176
 Options Text Blocks 176
 Repeat Blocks 386

And/Or 127

Annual Support Agreement 687

Apostrophe

in variables 140

Apply Wait 665

Archive IDB records 254

Array

Instant Database 218

ASA 687

'Ask' prompts

Automatic creation 509
 Examples 523
 General 499
 Recalling values 516
 Repeat 392
 Saving values 516

AskValue prompt 503

Assembly Options 64

Assign Prefix to Book 88

Augmenting Exiting IDB Record 194

AutoMask 262

Automatic

[Brackets] 122
 {Braces} 122

+[...] 122

<<Markers>> 122

Automatic Document Naming 546

Automatic Paragraph Numbering

General 324

Problems 325

Repairs 325

AutoReScan

Default Setting 212

Highlight 212

Temporary disable 212

AutoScan 207, 212

- B -

Baby steps 678

Backup 672

Balancing Markers 446

Beginner's Guide 26

Beyond assembly 25

Big Example 527

Blank Variables 208

Bold text 137

Book 69, 70

Actually a 'pointer' 21

Add first 35

Add New 76

Defined 5

Editing 447

Moving 654

Repointing 78

Shelving 76

Unshelving 81

Books

Rearrange in library 76

Rearranging 78

Bracketed variables 117

- C -

Calculate

Age 199

Date Difference 199

Date Math 199

Calculator 210

Calendar 224

Calls to <<files>> 104

- Caption
 - Example 399
- Captions 679
- Capture
 - Quick Picks 180, 182
- Capturing Data
 - Alternative 680
 - Instant Database 186
- Cascading
 - Alias *Lists* 468
 - Instant Database 473
 - Options Text 468
 - Simple Options 468
- Cascading logic 496
- Cascading Options
 - Chart 472
- Case (ALLCAPS, etc.) 137
- Case (Logic prompt) 518
- Chart
 - Cascading Options 472
- 'Cheat Sheet' 553
- Checklist for Upgrading System 38, 686
- Checklist of Clauses 422
- Clause Conversion 318
- Clause Name Table 66
- Clause names
 - Hidden text 65
- Clause Package 58
- Clause Selection Screen
 - Additional Assembly Options 64
 - Compare with Clause Sets 632
 - Compare with DropDown Lists 632
 - Double Click Tools 63
 - Elements 52
 - Freestyle Display 51
 - Repeat Function 395
 - Show All 57
 - Widen 59
- Clause Sets
 - Compare with Clause Selection Screen 632
 - Creating Automatically 407
 - Creating Manually 411
 - Editing 413
 - Expand 62
 - Expanding 415
 - from DropDown Lists 409
 - General 406
 - Issues 416
 - New page marker 416
 - Package 58
 - Using 414
- Clauses 306
 - .doc vs. .dot 308
 - Add manually 311
- Clearing
 - Styles 322
- Clipboard
 - DropDown Lists 364
- Clippy 26
- Collections
 - DropDown Lists 369
- Color Coding 447
- Columns
 - Deleting empty 110
- Comma separated values 254
- Comma usage 212
- Commas
 - IDB lists 203
 - Lists 203
- Comments
 - Adding 670
 - Purpose 670
- Common Profiles Path 557, 557
- Comparative Approaches 622
- Compare
 - Bracketed vs. Non-bracketed variables 627
 - Clause Name Table vs. No Table 66
 - Clause Selection Screen vs. Clause Sets 632
 - Clause Selection Screen vs. DropDown Lists 632
 - Clauses vs. Templates 622
 - DocManagement vs. DocAssembly 24
 - Folders vs. Glossaries 634
 - Glossary vs. Folder 75
 - GotForms? vs. InstantDatabase 633
 - 'Highlight & Add' vs. Direct Add 638
 - IDB Mask vs. IDB Scan 633
 - M/C variables vs. M/C *Lists* 627
 - Mouseless Assembly vs. Everything else 633
 - 'Optional' text vs. Clause Selection Screen 630
 - 'Optional' text vs. Individual Clauses 631
 - 'Optional' text vs. Variables 628
 - 'Options' text vs. Multiple choice variables 630
 - Pathagoras vs. Other DocAssembly Programs 665
 - Prefix/Suffix vs. 'Full Names' 625
 - Quick Picks (MS Word) 669

- Compare
 - SuperBooks vs. other locations 639
 - Compatibility
 - WordPerfect® 605
 - Worldox® 604
 - Computers
 - Adding 684
 - Replacing 684
 - Concatenation
 - In-line 237
 - Instant Database values 237
 - Conditional Text 141
 - Connectors 578
 - IDB lists 203
 - Lists 203
 - Consistency
 - Variables, DropDown List 128
 - Contact Us 690
 - 'Containers'
 - Glossary vs. Folder 70
 - Context Menu 440
 - Conversion
 - Simple to Advanced 20
 - Convert
 - {Simple} to <<*Optional*>> 168
 - Folder to Glossary 318
 - Glossary to Folder 318
 - Convert variables
 - HotDocs® 608
 - Converting
 - WordPerfect to Word 319
 - CPP 557
 - Create Options Assistant 155
 - Create Record
 - via Excel 294
 - via External Database 297
 - Create Variables Wizard 122
 - csv database links 479
 - Currency Function (IDB) 239
 - Customer Service 690
- D -**
- DAB (double angle brackets) 104
 - Data 192
 - Capturing Data 680
 - IDB (statistical) 254
 - Printing IDB, multiple records 254
 - Data Collection Screen 260
 - Data entry 186, 260
 - Data Records
 - Augmenting Existing Record 194
 - Creating 192
 - Location 192
 - Using 192
 - Database 254
 - Database, External
 - Create Record 297
 - See 'External Database' 286
 - Date
 - Extended 224
 - Date Math (Instant Database) 199
 - Date Math Features 224
 - Date, validation 198
 - Debugging
 - Balancing Markers 174
 - DropDown Lists 377
 - Editing 174
 - Glossary 74
 - Source Clauses 446
 - Structure Checker 173
 - Default Settings 656
 - Definitions 5
 - Book 68
 - Container 70
 - Library 68
 - SuperBooks 90
 - Delay Processing 665
 - Delete/Retain variables 208
 - Deleting a document
 - via SaveSmart 544
 - Design-a-Form (web based intake form) 297
 - Document Assembly 42
 - Checklist of Clauses 422
 - Clause Sets 406
 - Compare CSS with Clause Sets 632
 - Compare CSS with DropDown Lists 632
 - Freestyle Display 51
 - Insert clauses as separate documents 64
 - Insert Page Breaks 64
 - Into tables 110
 - Linear vs. Stacked 61
 - List of Clause Names 422
 - Margin Assembly 426
 - Options 64
 - Package 58

- Document Assembly 42
 - Paragraph Assembly 43
 - 'Pathagorizing' 116
 - Settings 86
 - Stacked vs. Linear 61
 - Tag along clauses 112
 - Techniques 356
 - Template Assembly 45
 - Within tables 110
 - WordPerfect® 605
- Document Assembly Issues
 - Access to source clauses 67
 - Ordering 67
 - Spacing between lines 67
- Document calls 104
- Document Dis-assembly 330
 - Highlight & Add! 334
 - SaveSmart 544
 - Term Works! 311, 334
- Document Logic 492
 - <<*If* prompt 506
 - <<*Set* prompt 508
- Document Management 534
 - SmartPaths 536
 - SuperSmartPaths 537
- Document Package 58
- Document Search 593
- Documents
 - as variables 197
- Don't save variable 216
- Double Angle Bracket Functions 104
- Double Click Tools 63
- Drag and Drop 131
- DropDown List
 - Instant Database 129
- DropDown Lists 356
 - Adding Content 359
 - Clipboard 364
 - Collections 369
 - Community 375
 - Compare wth Clause Selection Screen 632
 - Controls 364
 - Creating 'Free Hand' 358
 - Creating from Book 357
 - Creating from Clause Selection Screen 357
 - Debugging 377
 - Environments 371
 - Excel 614

- Insert item from list 362
- Insert Name 409
- Open Document for editing 364
- Panel 364
- Preview 364
- Repointing 360
- Special 375
- Toggle buttons 364
- Using 362
- Variables 372
- WordPerfect® 605
- Dump Demos 574

- E -

- Easy as Pi
 - Alias *Lists* 465
 - Glossary 315
- Easy as Pie
 - IDB Records 254
- Editing
 - and Testing 160
 - Balancing Markers 446
 - Clause Sets 413
 - Format Painter 452
 - Global Editing Tools 454
 - Source Clauses 447
 - Tips 452
 - Tools 446
 - Undo 452
- Embed Alias Table 488
- Embed Variables 206
- Emphasis 137
- End User Settings 575
 - As security option 660
- Endless Loop 148
- 'Enter' character 71
- Environments
 - DropDown Lists 371
- Equivalency function
 - Multiple Actors or possibilities 231
 - Single equivalency 230
- Error Messages 690
- Excel
 - Create Record 294
 - Dropdown Lists 614
 - Instant Database 616
- Excel Assembly 57

Exceptions
 Position Rule (Repeat/Options) 392
 Exclude Variable
 IDB 216
 Expand Clause Sets 62
 External Databases
 Connections to 286
 csv 286
 Definitions 286
 Masks 260
 Setup 287
 tsv 286

- F -

File Locations 578
 Discussion 657
 Instant Database 212, 250
 MultiChoice *Lists* 250
 Filter 539
 Find & Replace 448
 Find and Replace
 General 448
 Folder
 Adding Terms to 311
 Convert to Glossary 318
 Folder of Terms
 Folder Equals Book 69
 Folders
 as variables 197
 Font 137
 Footers
 Changing in Template 98
 Forms Package 58
 Fractions
 Spell out 242
 within options blocks 146
 Freestyle Display
 Clause Selection Screen 51

- G -

Getting Started 30
 Global Editing Tools 454
 Global Search & Replace 448
 Glossary 70
 Adding Terms to 313

Changing term name 71
 Convert to Folder 318
 Debugging 71, 74
 From Painted Document 316
 Line endings 71
 Pilcrow 71
 Quick Glossary 314
 Technical 'stuff' 71
 GotForms?
 General 348
 Minimized Screen 349
 Underlines as variables 351
 When to use 352
 GroupNames
 Simple Optional 164
 Groups
 and Alias *Lists* 476
 Options/Optional Text 151
 Pronouns 119
 Repeat Function 384
 Variables 119

- H -

Hard value 162
 Headers
 Changing in Template 98
 Repeating tables with . . . 387
 Headers and Footers,
 Scan for variables in 188
 Help 26, 690
 Help System Search 553
 Hidden text 65
 Highlight & Add!
 Term Works 311
 Highlight variables
 Automatic 212
 Manual 174
 Hopeless imbalance 148
 HotDocs
 Comparison 665
 Converting 135
 HotDocs®
 Convert variables from 608
 HotKeys
 <Alt-D> 590, 591
 <Alt-G> 590
 <Alt-N> 590

- HotKeys
 - <Alt-Q> 590
 - <Alt-S> 591
 - <Alt-V> 591
- HotWords 590
- Hover Over Text
 - Optional blocks 144
 - Options blocks 146
- Hunt Path 95
- | -**
- If
 - Multiple comparators 496
- If . . . Then 506
- If/Then
 - Logic Editor 510
- If/Then Logic
 - Elements 531
 - Examples 531
 - Structure 531
- Image Assembly 378
- Incrementing variables 383
 - Repeat function 383
 - Suppress incrementing 383
- INI settings 579
- In-line
 - Concatenation 230
 - Math 227
- Insert Name
 - via DropDown Lists 409
- Insert text
 - <Alt-G> 94
 - Clause Selection Screen 60
 - DropDown Lists 362
- Insert/New Doc toggle
 - DropDown Lists 364
- Inserting Variables
 - Drag and drop 131
 - DropDown List 129
- Installation 30
- Instant Database 208, 212
 - <\$>Currency Function 239
 - <<Filename>> Function 244
 - <<Folder>> function 228
 - <F>ormat Function 242
 - <P>aragraph Function 245
 - <S>pell-out Function 238
- Adding Individual Records (external source) 297
- Advanced Array 218
- Alphabetize Variables 210
- Archiving Records 254
- Augmenting Existing Record 194
- AutoReScan 212
- AutoScan 207, 212
- Calculator 210
- Calendar 224
- Cascading 473
- Concatenation 237
- Date Features 224
- Date Math 199, 224
- Delete/Retain variables 208
- Double angle bracket functions 244
- DropDown List 129
- Equivalency function 230
- Excel data 616
- Find and Replace Tool 217
- Input Form Masks 262
- Location of Records 250
- Mail To: Function 246
- Making the mask more meaningful 267
- Math 199
- Math & Date Math (in-line) 227
- Miscellaneous Features 208
- Personalize 212
- Pointing to Records 250
- Power Tools 210
- Print Data (multiple records) 254
- Resize (wider/taller) 210
- Save Variable Names 206
- Saving records 250
- Screen Shots 220
- Settings & Tools 212
- Sex Change Mask 217
- Sharing Records 254
- Sort Variables 210
- Table (statistical) 254
- Titled variables 121
- Tree Service 250
- URL: Function 247
- Validation of values 198
- Instant Database Masks
 - Assign Mask to Book 269
 - DropDown List 129
 - Location 250

Instant Database Masks
 Organizing the mask 268
 Page/Section titles 268
 Passwording 271
 Instant Database Records
 Location 250
 Instant Database Screen
 Basics 186
 Elements 186
 Tour 186
 Utilities 186
 Width 186
 Instant Interview 514
 Intake Form
 Client supplied data 274
 Excel 283, 294
 External, Web based 297
 Fancy (multi-part) 274
 Fancy, Scraping Data 278
 Fancy, Transmitting Form 281
 IDB Masks, creating 262
 IDB Masks, general 260
 Simple (Two Column) Table 273
 Integrity Checker 173
 Intelligent A 127
 Interview Wizard
 Limitations 513
 Print 517
 Introduction 2
 Italic 137

- J -

Joining Tables 110

- K -

KISS (keep it simple) 678

- L -

Length of variables 212
 Libraries & Books
 Concepts and samples 46
 Environments 371
 Screen 47
 Tour 47

Libraries & Books screen
 Adding Clauses 308
 Library
 Create New 82
 Defined 5
 Deleting 85
 Library, Add First 32
 Line Endings
 Glossary 71
 Linear Assembly 61
 Links 479
 List of Clause Names 422
 Lists
 Connectors 203, 578
 Default connectors 578
 Default separators 578
 Separators 203, 578
 Little Checkboxes
 Instant Database 218
 Locating IDB records 250
 Location
 Alias *Lists* 478
 Instant Database Records 250
 Pointing/Repoining 250
 Sharing 250
 Locking
 Protecting document against editing 661
 Logic
 Advanced (Interviews) 498
 Assistant 510
 Cascading 496
 Case 518
 Editor 510
 General Discussion 492
 If/Then 510, 531
 Intermediate 495
 Simple 493
 Logic expressions 506
 Logic Math 506

- M -

Mail Merge 300
 MailTo 246
 Main Menu 553
 Manually add clauses 311
 Margin Assembly 426
 Mask Maker 265

Masks 260
 Masks, Drag and Drop Values 131
 Math
 Age 227
 Calculate Age 199
 Date difference 199
 Date Math 199
 In-line 227
 Math (In-line) 227
 Matter 192
 Matter Records
 Augmenting Existing Record 194
 Creating 192
 Definition 192
 Location 192
 Using 192
 Matter Records (IDB)
 Locating 250
 Sharing 250
 Tree Service 250
 Messages 563
 Messaging 563
 Microsoft Assistant 26
 Misc
 Navigation 655
 Modifying variable names 117
 Mouseless Assembly 94
 Mouseless Navigation 550
 Moving books 654
 MultiChoice *Lists*
 Creating 460
 Definition 458
 From external data 479
 Virtual *List* 484
 Multiple choice text blocks
 Robust options 146
 Robust, syntax 152
 Simple Options 164
 Multiple Comparators 506
 Multiple Documents
 New Doc marker 416
 MyButtons 180, 182
 Search & Replace 448
 MyPathogoras Ribbon 610
 MyPicks 180, 182

- N -

Name and Subject Editor 428
 Names 60
 Changing 428
 Naming
 Automatic 546
 Naming rules 320
 Navigating 538, 598
 Navigation 598
 Navigation via textboxes 655
 Negative Optional 160
 A/K/A list 127
 Robust 144
 Simple Optional 171
 Nested
 Variables 135
 Nesting
 Alias *Lists* 468
 Maximum 148
 Options/Optional Text 148, 150
 Repeats 398
 Variables 150
 Network
 Enabling 557
 Messages 563
 Upgrading Pathogoras 564
 Network features 560
 Network Operations 556
 Synchronizing 566
 Use Local Files 566
 Networking
 Instant Database files 563
 Networking issues
 Instant Database 560
 Neutering 116, 131
 Never use original text 42
 New Computer 684
 New Doc marker
 Separate Documents 416
 New page marker
 Clause Sets 416
 Margin Assembly 416
 New Users 684
 Numbering 324
 Numbers
 Spell out 238

- O -

- One-Page
 - 'Optional' text 176
 - 'Options' text 176
 - Quick Start 696
 - Variables 176
- Open Folders 598
- Optional
 - General 141
- Optional selection by value 162
- Optional Text
 - {Simple} 164
 - Advanced 144
 - Anatomy 176
 - 'Ask' prompt 499
 - Negative Optional 144
 - Structure Checker 173
 - Summary 176
 - Testing 160
- Optional Wizard 155
- Options
 - Checkboxes 148
 - Fed by Repeat value 392
 - General 141
 - Radio buttons 148
 - Restrict to single choice 148
 - Tables and Rows 154
- Options and Tables 154
- Options Text
 - {Simple} 164
 - Anatomy 176
 - 'Ask' prompt 499
 - Prompts 146
 - Questions 146
 - Summary 176
 - Testing 160
- Options Wizard 155
- Options/Optional Text
 - Anatomy 152
 - Creating 155
 - General 141
 - Groups 151
 - Headers/Footers 157
 - Nesting 148
 - Nesting Variables 150
 - Suppress processing 158

- Syntax 152
- Order of Search 95
- Outlook
 - Alias *Lists* 479
 - Virtual *List* 484

- P -

- 'Package' of Forms 7, 58
- Paint
 - A/K/A 127
 - And/Or 127
 - Unpaint 127
- Paint Markers 447
- Pairing Table 299
- Paragraph Assembly 43
- Paragraph Numbering 324
 - Problems 325
 - Repairs 325
- Paragraph numbering (optional) 672
- Passwords 660
- Pathagoras Toolbar 654
- 'Pathagorizing' 37, 116, 131
- PathSmart 534
 - Settings Screen 549
 - SmartPaths 536
 - SuperSmartPaths 537
- PathSmart Screen
 - Filter 539
 - Range 539
- PDF Assembly 378
- Percent Function 241
- Personalize
 - Instant Database 212
- Phone number, validation 198
- Picture Assembly 57
- Pilcrow 71
- Plain Text
 - Advantages 23
 - Multiple Choice Variables 118
 - Reference to database 479
 - Reference to Outlook Contacts 479
 - Simple Variables 117
- 'Plain Vanilla' numbering 672
- Pointing
 - Book 78
 - Concept (general) 21
 - DropDown Lists 360

Pointing
 Instant Database Records 250
 Pointing to
 Alias *Lists* 478
 Position 1 Book 95
 Power Tools (IDB) 210
 PowerPoint 547
 Prefix Registration 424
 Prefix Table 420
 Prefix/Suffix Naming
 How it works 425
 Margin Assembly 426
 Modifying 424
 Prefix Table 424
 Printout 422
 Registering Prefix 424
 The 'Convention' 420
 Viewing 424
 What it is 420
 Why use it. 421
 Prepare folders 31
 Preserve Variables 206
 Preview
 DropDown Lists 364
 Preview Clauses 62
 Printout
 Checklist of Clauses 422
 List of Clause Names 422
 SmartPaths 539
 Problems
 Automatic Paragraph Numbering 325
 Process
 All open documents 218
 Options and Optional Text 160
 Processing vs. editing source text 17
 Processing vs. personalizing 17
 Testing 160
 'The' process 17
 Profiles 534
 Prompts
 Optional Text 144
 Options Text 146
 Pronouns 119
 Protection
 Protecting document against editing 661

- Q -

Quick Access Toolbar 610
 Quick Glossary 314
 Quick Parts (MS Word)
 Comparison 669
 Conversion 669
 Listing 669
 Quick Picks 180, 182
 Quick Set
 Book 77
 SmartPath 549
 Quick Start
 One-Page summary 696
 QuickLinks 592
 Creating 550
 Editing 550
 Using 550
 Quotation Marks
 in variables 140

- R -

Radio buttons 148
 Range 539
 Ranges
 vs. Positional 503
 Register Prefix 424
 Registry Settings
 Explanation of Settings 581
 Sharing 579
 Remote Assistance 574
 Renaming Documents 428
 Renewal
 Annual Support 687
 Repeat
 and Options 392
 Repeat Function
 Alternatives 395
 'Ask' 392
 Clause Selection Screen 395
 Connectors ('and'/'or') 386
 General 382
 Groups 384
 Incrementing variables 383
 Lists 386

- Repeat Function
 - Nesting 398
 - Restriction 398
 - Return repeat value 389
 - Rows in Tables 387
 - Settings 391
 - Simple 382
 - Tables 387
 - Testing 398
 - Replace text 448
 - via Instant Database 451
 - Replacing Computer 684
 - Repointing
 - Alias *Lists* 478
 - Book 78
 - DropDown Lists 360
 - Instant Database Records 250
 - 'Pointer' concept 21
 - Resize
 - Clause Selection Screen 59
 - IDB Screen 210
 - Restore 672
 - Right Click Menu 440
 - Rows
 - Deleting empty 110
 - Rules, naming 320
 - 'Running' Pathagoras 30
- S -**
- Samples
 - *Aliases* and MultiChoice Lists 458
 - *Ask* Table 529
 - <<*Ask*>> prompts 501
 - AskRepeat with Options 392
 - AskValue 503
 - Captions 679
 - Concatenation 237
 - Equivalency function 230
 - Equivalency function and *Actors* 231
 - Instant Database 199
 - Intake Form ('fancy') 281
 - Intake Form (Simple Two Column) 273
 - Interview 527
 - Libraries & Books 46
 - Math and Date Math 199
 - Operators 199
 - Options, simple/complex mix 170
 - Options/Optional Text 152
 - Pleading captions 399
 - Quick Picks 180, 182
 - Repeat 399
 - Repeat with Options 392
 - Repeat, connectors 386
 - Repeat, Groups 384
 - Repeat, return value 389
 - Repeat, Tables and Rows 387
 - Repeats, Incrementing variables 383
 - Signature blocks 679
 - Simple 199
 - Simple 'Repeat' block 382
 - Will 527
 - Sanitizing Document 122, 126
 - Save Variable Names 206
 - SaveSmart 540
 - Automatic Document Naming 546
 - Deleting a document 544
 - Document dis-assembly 544
 - 'More>>' controls 544
 - Moving a document 543
 - Proposing document name 546
 - Save to PowerPoint 547
 - Saving a document 542
 - Scan for Variables 188
 - Screen Shots
 - Clause Selection Screen 52
 - Clause Selection Screen (Additional Assembly Options) 64
 - Document Assembly Settings Screen 86
 - Got Forms? 350
 - Instant Database 220
 - Libraries & Books 46
 - Networking 557
 - Word 2007+ 610
 - Search & Replace 451
 - General 448
 - Global scope 448, 451
 - via Instant Database 451
 - Search Function 593
 - Search Help System 553
 - Search Order Rules 95
 - Security
 - Document Security 661
 - System Security 660
 - Self-contained documents. 112
 - Separators 578

- Separators 578
 - / (negative optional--robust) 144
 - / (negative optional--simple) 171
 - /ANDOR (simple options) 164
 - /OR 148
 - /OR (simple options) 164
 - Lists 203
 - Multiple Choice 148
 - Virtual *List* 484
- Series
 - IDB 203
- Settings 21, 570
 - 'All Settings' Screen 576
 - End user 575
 - Full Access 575
 - 'Repeat' Function 391
- Sex Change Mask
 - IDB 217
- Share IDB Masks 250
- Share IDB Records 250
- Sharing Alias *Lists* 489
- Sharing Data
 - Instant Database 254
- Sharing Registry Settings 579
- Shelving New Books 76
- Show All 57
- Signature blocks 679
- Simple Options
 - General 164
 - Groupnames 164
- Simple Options Blocks
 - Compare 167
 - Convert to <<*Optional*>> 168
 - Enabling/Disabling 169
- Simple to Advanced 20
- SmartPath
 - Adding 548
 - Adding via Settings Screen 549
 - Quick Set 549
 - Removing 549
 - Settings Screen 549
- SmartPaths 534, 536
 - Printout 539
- Social Security Number, validation 198
- Sort
 - IDB screen 210
 - Instant Database Variables 210, 268
- Sort Order
 - Names 428
- Source Clauses 306
 - Balancing Markers 446
 - Editing 447
- Spell out
 - Currency 239
 - Fractions 242
 - Numbers 238
- Stacked Assembly 61
- 'StartHere'
 - Page 100
 - Point 100
 - Section 100
- Structure
 - Options/Optional Text 152
- Structure Checker 173
 - Options and Optional Blocks 160
- Style/Caption (legal kind)
 - Example 399
- Styles
 - Alt-Q 322
 - Alt-Space 322
 - Restoring 322
- Subject and Name Editor 428
- Subjects 60
 - Adding 320, 428
 - Changing 428
 - Editing 320
- Summary
 - Optional Text Setup 176
 - Variable Setup 176
- SuperBooks 90
 - Adding Terms 92
 - SuperFolder 90
 - SuperGlossary 91
- SuperSmartPaths 537
- Support
 - Annual Support Agreement 687
 - Non-Technical 692
 - Technical 690
- Suppress processing
 - DropDown Lists 364
 - Options Text 158
- Synchronizing 565
- Syntax 180, 182
 - Administrative text (options blocks) 162
 - Options/Optional Text 152

- T -

Tables

- Document assembly within 110
- Erasing empty rows and columns 110
- IDB (statistical) 254
- Joining 110
- Options 154
- Repeat Rows 387

Tag Along Clauses 112

Technical Support 690

Template

- Exempting use 102

Template Assembly 45

Templates 96

- Assign to Book 101
- Assign to Source Document 102
- Attaching to book 98
- Changing headers/footers 98
- Creating 98
- Defined 96
- Exempting from external template 98
- Screen 97
- 'StartHere' Point 100

Temporary variable 216

Term Works! 311

- Document Disassembly 334
- Highlight & Add! 334
- Select text 335
- Show text 335, 343

Testing

- Options and Optional Text 160

Textboxes

- Scan for variables in 188

Tips and Tricks

- Baby Steps 678
- Captions and Signature blocks 679
- Capturing Data 680
- KISS 678

Titles (variables) 121

Toolbar

- End User 575
- Four-element 654
- Two-element 655

Tree Service

- Clause Selection Screen 106
- Instant Database Records 250

Typing expander 592

Typographical errors 65

- U -

Underlines as variables 351

Uninstall 694

Unshelving Books 81

Unused Variables

- Delete 212

Upgrade Checklist 38, 686

Upgrading Pathagoras 573

- General Issues 684

- Network installations 564

- Standard 26, 684

URL 247

Use Local Files 566

Utilities/Settings 572

- Current Settings 572

- File Locations 578

- V -

Validation

- Date, Phone number and SSN 198

Value

- Return repeat value 389

Values

- Analyze 503

Variable Creation Wizard 122

Variables 188

- Alphabetize 174
- Alphebetizing 268
- Apostrophe 140
- as Documents 197
- as Folders 197
- Auto-count the brackets 174
- Changing names 176
- Color the brackets 174
- Completing 191
- Consistency 128
- Debug 174
- Definition 117
- Delete if blank 208
- Don't save 216
- Drag and drop 131
- DropDown List 129

Variables 188
 Embedding 206
 Groups 119
 Headers and Footers 188
 Highlight 174, 210
 Incrementing w/ repeats 383
 Length 140, 212
 Modifying names 176
 Multiple Choice 118
 Nested 135
 Number of 195
 Pronouns 119
 Quotation Marks 140
 Replacing 192
 Reusing 128
 Scan for, 188
 Search & Replace 448, 451
 Simple Variables 117
 Sorting 268
 Special 197
 Temporary 216
 Testing 160
 Textboxes 188
 Titled variables 121
 Underlines (GotForms) 351
 Unused 212
Variables List
 DropDown 372
Virtual *List* 484

- W -

Web based Intake Form 297
Whole system search 593
Widen Screen
 Clause Selection 59
 IDB 210
Wizards
 Optional 155
 Options 155
Word 2007+
 MyPathagaoras ribbon 610
 Quick Access Toolbar 610
WordPerfect® 318, 605
 Assembly 378
 Conversion to Word 319
 DropDown Lists 378
Worldox® 604

Pathagoras prides itself on providing prompt, useful and personal customer service. While we hope that this Manual and the other instructional materials are helpful, you can still count on the 'personal touch' of Pathagoras' customer service as a 'first line' of help as well.

We truly enjoy hearing from our customers and potential customers. While we cannot promise that you will never receive a voice-messaging service if you call us, more likely you will receive a live person at the other end. If you do get our voice mail, just leave a message. We will call you back promptly.

Contact information is spread across as many places as we could find so that you do not have to hunt for an email address or telephone number, and we repeat it here. Let us know if we can ever be of service.

Pathagoras
Innovative Software Products of VA, LLC
Roy Lasris, President
www.pathagoras.com
info@pathagoras.com
tel: 866-PATHAGOras (1-866-728-4246) (tollfree)
tel2:+1 (757) 877-2244 (USA)
117 Chisman Landing
Seaford, VA 23696 USA